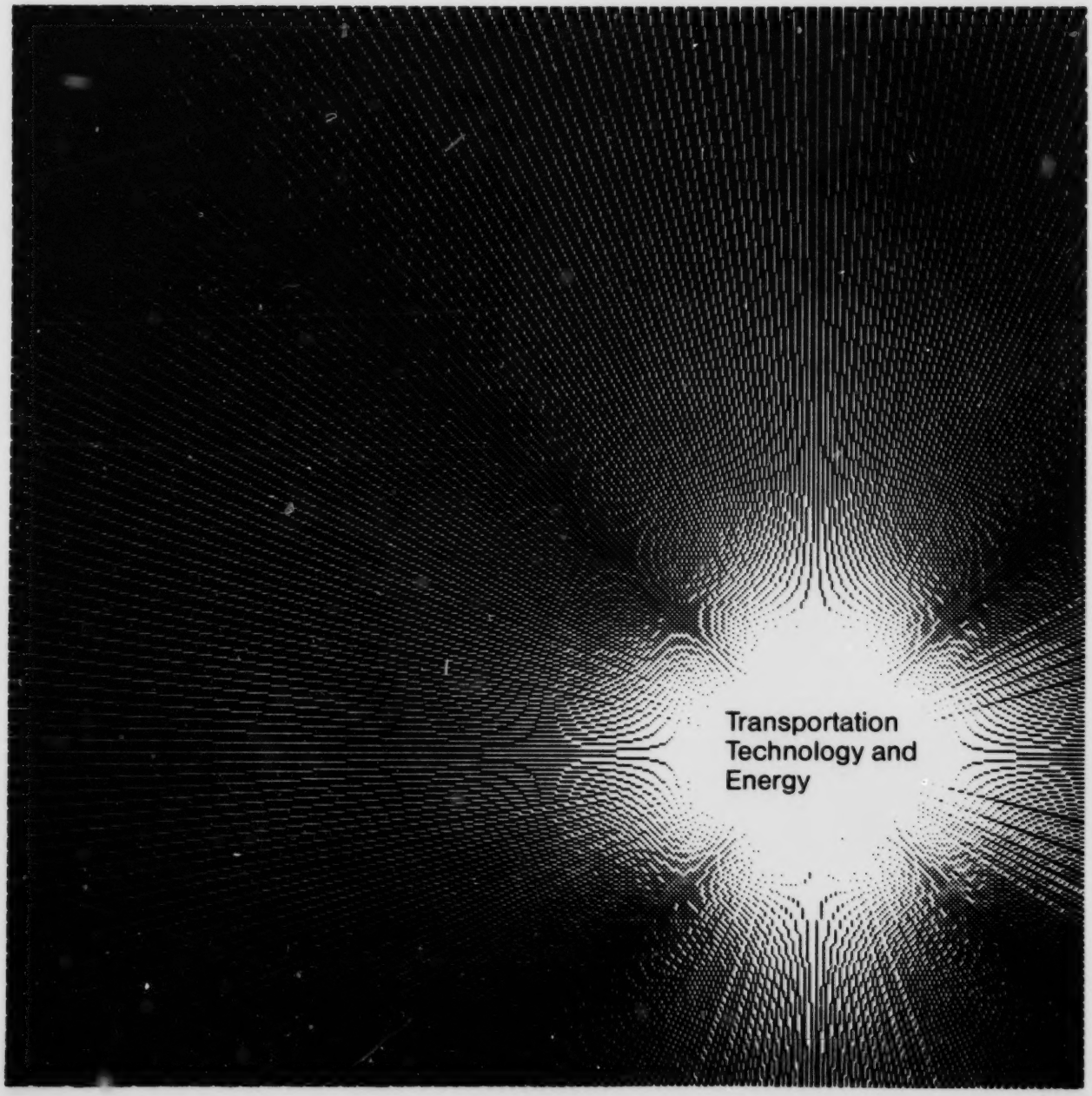
Ontario

Ministry of
Transportation and
Communications

Transportation
Technology and
Energy Branch

M-84-01

Operator/Computer Systems Dialogue in an Interactive System



M-84-01

Operator/Computer Systems Dialogue in an Interactive System

Abt Associates of Canada
E. L. Weinstein
J. Tombaugh
K. Grant

Project Manager
L. V. Clifford
Human Factors Section

Prepared for:
The Operations Management System
Research Steering Committee

L. Brant	A. Hawkins
D. Duncan	H. Rooke
P. Ginn	N. Varmazis

Published by:
The Transportation Technology and Energy Branch
Ontario Ministry of Transportation and Communications
Hon. James W. Snow, Minister
H. F. Gilbert, Deputy Minister

Published without prejudice
as to the application of the findings.
Crown copyright reserved; however, this
document may be reproduced for non-commercial
purposes with attribution to the Ministry.

For additional copies write:
The Editor, Technical Publications
Ministry of Transportation and Communications
1201 Wilson Avenue
Downsview, Ontario
Canada M3M 1J8

March 1984

Technical Report Documentation Page

MTC Report Number M-84-01	ISBN Number:	Other Document Number:	Date: Mar. 1984
Title: Operator/Computer Systems Dialogue in an Interactive System			
Authors (name, title and/or organisation): Abt Associates of Canada			
Published by: The Transportation Technology and Energy Branch, MTC		Client Group: The Operations Management System, Research Steering Committee, MTC	
Participating Agencies: Human Factors Section, MTC			

Abstract:
The study assesses the effectiveness and efficiency of the communication between the computer and the user of interactive systems by means of a case study and a review of relevant literature. The Operations Management System (OMS) of the Ministry of Transportation and Communications (MTC) was used as a study case. Advantages, disadvantages, problems and probable solutions and guidelines for future interactive systems were pointed out, discussed and explored.

Key Words: Operator/Computer Systems Dialogue, interactive system

Comments:

Distribution: by request

Copyright Status: Crown copyright

CONTENTS

	PAGE
EXECUTIVE SUMMARY	v
ACKNOWLEDGEMENT	xi
1/ INTRODUCTION	1
1.1/ Background to Study	1
1.2/ Study Methodology	1
1.3/ Report Organization	3
2/ CASE STUDY OF THE OMS SYSTEM	5
2.1/ What You are Doing Right	5
2.1.1/ Design	5
2.1.2/ Management	6
2.2/ Problem Identification and Impact	6
2.2.1/ Inconsistency	7
2.2.2/ Redundancy	8
2.2.3/ Error-Handling.	8
2.2.4/ Other Inefficiencies	9
2.3/ Solutions	9
2.3.1/ Changes in Data Entry	10
2.3.2/ Improve Operator Flexibility	11
2.3.3/ Add Files	12
2.3.4/ Other Issues Requiring Attention.	12
2.3.5/ Summary of Major Recommendations.	13
3/ GUIDELINES FOR FUTURE SYSTEMS	15
3.1/ The General Principles Underlying Guidelines	15
3.2/ More Specific Suggestions for Guidelines.	17
3.2.1/ Form Design.	17
3.2.2/ Screen Design.	19
3.2.3/ The Data Entry Process	21
3.2.4/ Menu Design and Operation.	25

PAGE

3.2.5/ Response Time and Display Rates. 26
3.2.6/ Feedback and Error Messages. 26
3.2.7/ On-line Help Systems 28
3.3/ The Need for User-Testing 30
3.3.1/ Review of Existing Systems 30
3.3.2/ Review During the Design Process 31
3.3.3/ The Use of "Prototype" Systems 31
3.4/ Literature Identifying Guidelines 32
3.4.1/ W.O. Galitz, "Handbook of Screen Format Design", Q.E.D. Information Sciences, Inc., 1982. 32
3.4.2/ W.O. Galitz, "Human Factors in Office Automation", Life Office Management Association, 1980 33
3.4.3/ S.L. Smith, "User-System Interface Design for Computer-Based Information System," A report prepared for the United States Air Force, April 1982 34
3.4.4/ M. Dean, "How a Computer Should Talk to People", I.B.M. Systems Journal, volume 21, No.4, pp. 424-453, 1982. 35
3.5/ Trends in Workstations. 36
3.5.1/ Distribution of Intelligence 36
3.5.2/ The Move Towards "Executive" or "Professional" Terminals. 37
3.5.3/ The Use of Multi-Function Workstations 37
3.5.4/ Changes in the Basic Design of the Workstation 38
3.5.5/ The Move Towards Portable Terminals. 38
3.5.6/ The Increased Focus on "Ergonomics". 39

REFERENCES 40

APPENDICES 42

LIST OF APPENDICES

PAGE

A/ Method Documents 42
B/ Description of the OMS System. 49
C/ System Problems. 51
D/ Solutions 70
E/ Summary of Problems and Solutions. 78

LIST OF TABLES

1/ Inconsistencies. 81
2/ Redundancies 82
3/ Error-Handling Problems. 83
4/ Problems Not Elsewhere Classified. 84
5/ Suggested Changes in Data Entry and Report 85
6/ Suggested Changes for Improving Operator Flexibility 87
7/ Other Suggested Changes. 88
8/ Solutions to Inconsistencies in Table 1. 89
9/ Solutions to Redundancies in Table 2 90
10/ Solutions to Error-Handling Problems in Table 3 91
11/ Solutions to the Unclassified Problems in Table 4 92

EXECUTIVE SUMMARY

1/ DESCRIPTION OF THE STUDY

The study assesses the effectiveness and efficiency of the communication between the computer and the user of interactive systems by means of a case study and a review of relevant literature. The Operations Management System (OMS) of the Ministry of Transportation and Communications (MTC) was used as a study case for starting to develop credible ergonomic guidelines for the information exchange between users and computers. Thus, the study made specific recommendations for improving OMS, but focussed on developing guidelines for future systems.

The study considered the OMS as a case study in developing guidelines for interactive systems. Thus, the study was restricted to the interactive part of OMS -- the remote sub-system. The remote sub-system is a series of distributed micro-processors located in a number of offices around the province. It is primarily a form-filling, menu-driven system used largely for data entry and local reports. The system has been implemented in different ways in different offices. In particular, offices were free to choose how many people would use the system and the ways in which they would get information out of the system.

The study is primarily qualitative in nature. After a literature review and some familiarization with the system, an initial round of observation and interviews with users led to a preliminary description of system problems. This was also derived from the careful examination of system features using a checklist developed by Smith (1982) for user-system interface features. Preliminary work was carried out at four sites.

Preliminary findings were discussed with the Project Advisory Committee and the second phase of the project was then begun. The second phase consisted of structured observation at an additional six sites. This was accompanied by brief interviews, an examination of errors, and some additional attempts to verify earlier findings.

The report resulting from this work briefly describes the strengths of the system. Its weaknesses are presented in greater detail. The report also suggests solutions to major problems on the current system, but without regard for their ease of implementation. Based on this work and our review of the literature, guidelines for similar future systems are presented. A few selected references are also given which identify additional guidelines. Finally, we provide some indication of the trends in workstations which may affect the design of future systems.

2/ ASSESSMENT OF THE CURRENT SYSTEM

As a data entry system, the OMS works quite well. In addition, users of the system generally like it. The system is widely accepted. There are a variety of reasons for the high degree of acceptance of the system. These reasons fall into two categories: design and management.

The decision to use the menu-driven, form-filling approach to data entry is probably the key to understanding user satisfaction with the system. The user can easily grasp the idea of filling in the screen. The consequences of their actions are immediately observable and most can be reversed. Thus, very little training time is required before any user can do useful work on this system. Routine data entry can be performed with very little instruction, particularly if a resource person is located nearby to answer questions about typical cases.

In addition, there are opportunities for interested users to increase their knowledge about the system. The combination of keying-in from meaningful forms and immediate feedback on the screen, helps the user to be aware of the meaning of the data being entered and the constraints associated with a certain form. Thus, the screens themselves and the ways in which special cases are best handled become familiar.

Another important design criteria which has been generally followed is an effort to build a consistent system. While we can identify some inconsistencies in the system, these are exceptions in a generally consistent system.

In addition to design, there are two aspects of management which foster user acceptance. First, each office was allowed to implement the system in the manner which best suited its own style and personnel. Second, the system experts have made themselves readily available to the users to answer questions and help solve their problems. This high availability and efforts to help the users with their problems reduces the frustration of trying to deal with uncertainties and difficulties in the system.

The major problems with the system can be organized into four groups: inconsistency, redundancy, error-handling, and other inefficiencies. Specific problems are detailed in the body and appendices of the report.

In additions, solutions to major problems are also identified. Solutions identified include:

- Provide free cursor movement and an overstrike capability for error-handling.
- All information in a batch should be correctable during data entry and MODIFY.
- Switch validation from field validation to line validation.
- Any information which must be entered on every line in a batch should be moved to the batch header and eliminated from the line.
- Add recommended local validations of operation/patrol/highway/crew and account number/cost centre.
- Increase the field sizes for hours and litres in the relevant screens.
- Allow addition and deletion of lines from a transaction in the MODIFY routine.
- Provide a facility for scanning all entries in a batch.

- Improve the documentation on the report generation system.

- Add vendor account files to replace names and addresses.

3/ GUIDELINES FOR THE FUTURE

Both general and specific guidelines were provided for developing future systems. The general guidelines emphasized five major directions:

- Improve efficiency -- minimize user effort during input and control actions, minimize errors, and minimize search times when looking at forms.

- Minimize the memory load on users.

- Make systems consistent -- the outcome of pressing any key should be consistently predictable.

- Maintain compatibility between source documents and information requested on a screen.

- Improve flexibility -- maximize the degree of user control over data entry, data display, error correction and sequence control.

In addition to these general guidelines, specific design guidelines are presented. For form design, there are a number of guidelines offered. These include:

- Information should be arranged in the order in which it is normally found.

- Items used more frequently should be near the beginning of the form.

- Items serving the same purpose should be grouped together.

- Important items in a group should be positioned in a prominent place.

- The organization of the data must make sense to the person providing or collecting the data on the forms.

- Forms should be designed so that users do not have to make decisions about changes or special cases while they enter data.

Guidelines for screen design largely follow those for form design. In addition, several added guidelines are offered:

- When entering data from a form, the screen design should parallel the form design.

- When entering data directly, entries should be aligned in columns, typically with one entry per row.

- When a form must be broken into several screens, it should be done in a way which keeps functional groupings of items together.

A number of guidelines are offered for the data entry process, but it is noted that different authors view this process differently. Consistent suggestions include:

- Appropriate defaults should be displayed in their appropriate fields on initial display of the screen.

- Do not use defaults for single column entries, but rather for longer fields.

- Users should manually tab from entry to entry, even when there is a fixed field format.

- Special functions should be available for skipping fields and moving the cursor for error correction.

- Fields should be entered into the system in complete logical groups, with entry triggered by an explicit user action.

- In data entry from forms, when typing ahead of the cursor's current position is supported, error-handling should take into account that users may be ahead of the cursor action.

In terms of menu design and operation, the same organizational considerations go into the design of menu items that go into the design of forms: natural sequencing, functional categories, frequency of use, and importance should all be considered. In addition, it is important to word the menu choices in a manner that makes the alternatives clear to the user. There should also be consistency in the way that menu items are selected.

Comments on response time and display rates are rather basic. Appropriate times, as indicated by the literature, differ with application. Perhaps the sole guiding principle is that the system should keep up with the user.

Suggestions for feedback and error messages are few. Users should always be given an indication that a valid command has been received and processed. When there are mistakes, auditory signals should be used only under very limited conditions. Error messages should be present on the screen until corrective action is taken and they should be clearly worded and avoid any possible misinterpretations. Finally, when a error is detected, the cursor should move to the error position automatically.

Finally, the report suggests the usefulness of an on-line help system.

For all of these guidelines, and many less important guidelines, the text presents a clear rationale for use. In addition, a brief description of four selected references which identify guidelines of use to system designers is provided, as well as a brief rationale for user testing. Finally, trends in workstations which could affect future systems design are briefly explored.

ACKNOWLEDGEMENT

We would like to thank Janace Pierce and Robert Rosenbaum for their work in developing this project.

We would also like to give special thanks to the members of the OMS Research Steering Committee for their participation. This participation shaped the direction of the project and was consistently of high quality. All of the Steering Committee members contributed to the project to an unusually large extent. Their enthusiasm and commitment has made the project successful.

In addition, we would like to thank the staff and operators at the many offices we visited for their cooperation. These offices include: Hamilton, Head Office Finances (Toronto), Kingston, London, Ottawa, Port Hope, Sault Ste. Marie, Stratford, Sudbury, and Toronto. In each office, we found people cooperative and interested in expressing their views about the OMS system. Without this cooperation, this report would not have been possible.

Finally, we would like to express our regrets at the passing of one of our Steering Committee members, Peter Geary.

1/ INTRODUCTION

1.1/ Background to Study

The terms of reference for this study were to assess the effectiveness and efficiency of the communications between the computer and the user of interactive systems by means of a case study and review of relevant literature. The Operations Management System (OMS) of the Ministry of Transportation and Communications was used as a study case in making a start toward "development of credible ergonomic guidelines for the information exchange between users and computers" (terms of reference). Thus, the study aimed to produce guidelines for future systems by focussing on the response of users to the current system and an independent assessment of system features.

More specifically, the study considered the following issues:

- effectiveness of the dialogue,
- response time acceptability,
- error detection system,
- problem solving,
- the operator's overall perception of the system.

The study was directly concerned with the interaction between the end user and the computer. Thus, other aspects of the system were not studied, even though these were also factors which affect the overall effectiveness of meeting the data processing needs of the Ministry. Some of these factors are data flow (transfer of data from source through intermediate clerical functions to the point of data entry), data base design, office layout and general environment, and central (host computer) data processing.

1.2/ Study Methodology

The methods used in this case study were different from the methods originally proposed. The original method consisted of the following phases:

- familiarization with the system,
- questionnaire design, distribution and analysis,

- in-depth interviews and observations at a small number of sites.

In addition, we planned to use a checklist developed by Smith (1982) to evaluate which aspects of the system require modification. It was thought that this might be a starting point for establishing an easy-to-use and practical index for system designers.

The familiarization phase was carried out according to plan. We learned about the system from reading manuals and visiting sites to use the system. In addition, managers and operators were interviewed to determine the nature of the questions to be included in the questionnaire. As a result of this experience, we prepared a preliminary report and recommended that the questionnaire not be used. There were several reasons for this request. The most important were:

- Users tended not to be able to think of problems with the system until they were using it. Thus, open-ended interviews while watching seemed a more productive method.

- Users could not really express the magnitude of the problems which they could remember, such as slow system response time. We needed to observe users in action to discover the magnitude of the problem.

- Two of the general complaints, redundant information being required on the forms and inconsistencies, could only be appropriately documented by becoming much more familiar with the forms.

All of these considerations argued for site visits rather than questionnaires.

Preliminary work also showed us that Smith's checklist was only helpful in initial assessment. Smith's checklist just identifies features which might be useful in a system -- without regard for the nature or purpose of a system. Thus, many of Smith's checklist items (e.g., graphic capabilities) were not appropriate to OMS. As well, many of the features were not clearly described. This made them difficult to assess. In fact, we found Smith's guidelines to be more useful than his checklist. Indeed, we found that some of Smith's guidelines, supplemented by

Galitz's (1980) guidelines on form and screen design, were very useful. Thus, we shifted our emphasis from using the list of features to using the guidelines.

For the reasons cited above, we proceeded directly to intensive observation and omitted the survey. Six sites were selected for visits. They represented a cross-section of large and small, regional district offices. Operators were observed while processing three different batches of information which were selected by the site. We watched for typing rate, number and types of errors and methods of correction. We also asked them to describe the errror correction procedures they used and to comment on changes they would like to see on the forms. The interview schedule used is provided in Appendix A.

We also noted where Smith guidelines applied and whether or not they were particularly relevant to the OMS. For example, we consider guidelines about form and screen consistency to be of high importance, since users spend most of their time filling forms. But, guidelines about setting up menus (e.g., most used items first) are not of great importance to this particular application in which menu use is limited and follows a simple pattern.

Finally, we looked at the kinds of errors which were occurring on the system as a further guide to the types of changes most needed in the system. As noted by Norman (1982), an examination of the types of errors made is a key to system design problems.

1.3/ Report Organization

Section 2 describes a case study based on the current system and Section 3 provides guidelines for future systems. Essentially, the purpose of Section 2 is to make recommendations, based on the case study, that will have implications for future systems. The section is divided into four subsections. The first subsection identifies those things which the users like and which should not be changed. The second subsection describes major problems with the system. The third subsection then describes some possible solutions. We would like to emphasize, however, that we have not investigated the feasibility of these solutions, nor can

we comment on the relative importance of these changes in respect to changes in the non-interactive parts of the OMS. It is possible that the interactive portion is in less need of attention than other aspects of the system and that many of the recommended changes will be given a low priority with respect to other needs.

Section 3 describes the guidelines that we consider important for future development of form-filling systems. We hope that they will be of help to future designers. Many of the recommendations cost nothing if adhered to from the beginning of the design effort -- so they are easily adapted.

The appendices, include:

- method documents,
- system description,
- detailed description of system problems,
- detailed description of recommended solutions,
- correspondence between problems and solutions.

2/ CASE STUDY OF THE OMS SYSTEM

2.1/ What You are Doing Right

While the nature of the objectives of this study required that we document its shortcomings in as much detail as possible, we wish to take this opportunity to place our section on "problems" into perspective. Basically, the OMS, as a data entry system, works well. In addition, users of the system generally like it.

We feel that we can identify some of the reasons for the high degree of acceptance of the system. These reasons fall into two categories: design and management.

2.1.1/ Design

The decision to use the menu-driven, form-filling approach to data entry is probably the key to understanding user satisfaction with the system. The user can easily grasp the idea of filling in the screen. The consequences of every action are immediately observable, and most can be reversed. Thus, very little training time is required before a new user can do useful work on this system. Routine data entry can be performed with very little instruction, particularly if a resource person is located nearby to answer questions about atypical cases.

In addition, there are opportunities for an interested user to increase his or her knowledge about the system. First, the combination of keying in from meaningful forms and immediate feedback on the screen, helps the user to be aware of the meaning of the data being entered and the constraints associated with a certain form. Thus, the screens themselves and the ways in which special cases are best handled become familiar. The user feels growing competence in performing the tasks and increased mastery of the system. Thus, while a quick start on the system is possible, a continuing positive attitude is maintained. Shneiderman (1982) has pointed out that most users respond positively to a wide range of "direct manipulation" programs, such as those in the OMS, because the object the user is working on (the form) is visible and directly manipulated or changed by the user's actions.

Another important design criterion which has been generally followed is an effort to build a consistent system. While we point out some inconsistencies on the OMS system, we feel that we are pointing out exceptions in a generally consistent system. Galitz (1980) among others, considers consistency to be one of the most important design criteria.

2.1.2/ Management

The acceptance of a computer system depends upon many features other than its technical quality. There are two aspects of management which we feel deserve positive comment in this regard, because we see them as important to the users of this system and because they are often not followed in system implementation.

First, each office was allowed to implement the system in the manner which best suited its own style and personnel. As mentioned earlier, some offices have data entry experts who do most of the keying, while others have several persons who do data entry as a small component of their tasks. Too often, systems are brought into an office along with an implementation scheme which has been centrally decided upon without regard for individual variations.

Second, the system experts have made themselves readily available to the users to attempt to answer questions and help solve their problems. This high availability and efforts to help the users with their problems reduces the frustration of trying to deal with uncertainties and difficulties.

2.2/ Problem Identification and Impact

There are many ways to classify the problems we have identified with the system. We have chosen to classify these problems into four broad categories:

- inconsistency,
- redundancy,
- error-handling,
- other inefficiencies.

Within each of these broad headings we have further sub-divided our findings into a number of finer categories. Within categories, we try to present the major type of impact created by the problem we have identified. The types of impact are:

- reduced efficiency of data input and correction,
- increased number of errors,
- shift of work from machine to manual operation,
- reduced operator satisfaction with the system.

2.2.1/ Inconsistency

Inconsistencies are presented in three groups: system, form, and report inconsistencies. System inconsistencies refer to problems encountered in menus, default options, and exit procedures. Form inconsistencies refer to differences in location of information on a source document as compared to the screen. Report inconsistency refers to problems in requesting reports and the kinds of information available from them.

In total, some 21 inconsistencies were identified (see Appendix C). Of these, 11 were system inconsistencies, six were form inconsistencies, and four were report inconsistencies.

The most significant of the system inconsistencies was a lack of a consistent method of exit from OMS routines.

For form inconsistencies, the most serious one is the lack of a sufficient number of digits for hours and litres of fuel on the maintenance management screen. Similar problems occur with equipment cost and garage work order screens.

The most important of the report inconsistencies is simply that the whole report request procedure confuses its users. They do not understand what they can expect when they make a particular report request. A further source of annoyance is that information keyed-in is sometimes not available to people in the local offices.

2.2.2/ Redundancy

Redundancy refers to system constraints which add unnecessary keystrokes to the operator's job. There are several types of redundancy that we have identified including:

- information common to all transactions in a batch which is entered for each transaction,
- long strings of information which could be replaced by a shortened code,
- repetition of information without any known purpose.

In total, nine major sources of redundancy are identified. Two of these are things which are of particularly high impact in terms of time consumed, errors produced, and operator annoyance. These two problems are:

- repetition of identical information in every transaction of a batch, specifically the effective date and the transaction type,
- typing of names and addresses.

2.2.3/ Error-Handling

Error-handling on the OMS is rudimentary. In particular, error correction during data entry is extremely limited. Some 17 error-handling problems can be identified (see Appendix C).

Among the many problems with error-handling, there are six problems which we consider to have a significant impact on the system. These most serious problems are:

- It is impossible to correct all information visible on the current screen.

- Backing-up to correct information cannot be done without loss of already entered information and retyping.

- You cannot add a line to or delete a line from a transaction.

- Some fields cannot be modified.

- Incompatibilities in operation type with patrol/highway/crew type result in rejected batches.

- Incompatibilities between account numbers and cost centres result in rejected batches.

2.2.4/ Other Inefficiencies

A number of other problems we identified do not fall conveniently into any of the three categories already discussed. These tend to be general complaints about how the system is working or specific complaints about some of the routines. We conceive of these issues in two groups: applications-oriented and system-oriented.

In total, we identified seven applications-oriented problems and 14 system-oriented problems (see Appendix C). The most serious of the applications-oriented problems is the lack of understanding of report generation. An understanding of this routine seems critical, but is clearly lacking. It is likely that reports are not being used to the full extent possible.

In terms of system-oriented problems, there are two which users consider particularly serious:
- the slowness of data entry,
- transmission problems.

2.3/ Solutions

In this section we briefly identify methods of dealing with the problems identified in Section 2.2. Solutions are divided into three broad categories:

- changes in data entry procedures,
- improved operator flexibility, and
- added files.

We would like to emphasize that we have not investigated the feasibility of our solutions, nor can we comment on the relative importance of these changes in respect to changes in the non-interactive parts of the OMS. It is possible that the interactive portion is in less need of attention than many other aspects of the system.

2.3.1/ Changes in Data Entry

There are five changes in data entry procedures which we consider imperative. First, information which is repeated on every line of a batch should be moved to the batch header. This includes such information as date, transaction type, and activity code. This should result in more efficient data entry and would remove one major source of operator dissatisfaction with the system. Second, validation should be done on entire lines of data, rather than on individual fields. That is, each field should still be validated; but all the fields on a line (or in a logical grouping) should be checked after the entire line (or grouping) is entered. This, combined with later suggestions for error correction, should make correction easier for users. As well, by reducing the number of switches between terminal and processor, the speed of data entry should be improved.

A third change of major importance, is to increase field sizes in some of the maintenance transactions. In particular, hours worked should be allowed to exceed 1000 and the litres of fuel consumed should be allowed to exceed 10 000. This would have a substantial impact on reducing errors and time of entry. It removes the need for notional transactions which are a major burden to users entering maintenance reports.

Finally, adding two local edits to the system would have a substantial impact in reducing batch rejections and thus would improve turnaround time for reports and reduce the need for manual intervention to correct errors. First, there should be a local compatibility check on the patrol/highway/crew type combination for each operation type (error

2161). Errors of this type occur in 24% of rejected batches. Thus, a compatibility check could decrease the number of rejected batches significantly. The second useful compatibility check is between cost centres and associated account numbers. This accounts for 15% of batch rejections and errors detected prior to transmission.

2.3.2/ Improve Operator Flexibility

In general, suggestions for improving operator flexibility are more difficult to implement than those in the preceding section. They tend to have wider ramifications on how operators can work with the system. Most of the suggestions arise out of problems in handling errors.

For error-handling, the single most important change to be made is to replace the backspace and erase procedure for correcting errors. Instead, the user should be able to move the cursor to any place on the screen without erasing the material between the position of the cursor, when an error is found, and the position of the error itself. Then, the user should be able to overstrike any characters to be changed. Subsequently, the user should be able to move the cursor back to its original position and continue entering without any penalties. Accompanying this free cursor movement and overstrike capability, the user should be able to alter everything in the batch. No field should be inaccessible. By increasing the ability to correct errors during data entry, less use of the MODIFY routine will be required.

Another recommendation with high impact is that the MODIFY routine allow users to add or delete any line from any transaction or form. It should not be necessary to remove a transaction entirely or add an entire new transaction to make this kind of alteration. This will improve the efficiency of data entry and modification considerably.

Error-handling will also be facilitated if there is a method for scanning the transaction in a batch.

2.3.3/ Add Files

The highest impact suggestion we can make is to add a vendor file to the system using telephone numbers to replace names and addresses of vendors which are commonly entered by users. The use of a vendor code, instead of names and addresses, could be optional. You might restrict vendor codes to province-wide entities like Bell Canada or Ontario Hydro. Alternatively, and this is our long-term suggestion, you could use it for all regularly invoiced corporations and for all expense accounts. Vendor code files would eliminate a large part of data entry time. They would also eliminate a major source of errors found prior to transmission and errors found in batch rejections. Aside from eliminating information which is repeated on every line, this is probably the second most common suggestion made by users.

The vendor code file is certainly the only high impact file that can be added. In addition, however, we would also suggest adding a local file containing information on reports which have been requested. This file would contain the report number, title, parameters requested, and the values of each parameter requested in the report. You could also make this file valuable by working it into a routine which displayed a quick schematic of what the report would look like. This approach might relieve the need for more elaborate documentation of the report generation system.

2.3.4/ Other Issues Requiring Attention

There are a variety of remaining issues which require attention (see Appendix E). Two of these are important. First, users are nearly unanimous in their desire for better documentation of the report generation systems. There have been some questions about the flexibility of the system, but we believe this may be due to a lack of understanding of how it works. Documentation of the report generation routine is a high priority. Note though, that the documentation should include many examples of what would appear from the set of parameters.

A second area requiring documentation is transmission problems. Most sites would like a basic manual indicating what to do about some commonly

occurring transmission problems. While they recognize that good support is available at headquarters, they would prefer to deal with their problems systematically and on-site.

2.3.5/ Summary of Major Recommendations

We have presented a number of recommendations in four different sections. Here we identify the recommendations which we believe would best respond to the concerns of your users. As well, these are recommendations which would have a major impact upon the efficiency of system operation. These recommendations include:

- Provide free cursor movement and an overstrike capability for error-handling.
- All information in a batch should be correctable during data entry and MODIFY.
- Switch validation from field validation to line validation.
- Any information which must be entered on every line in a batch should be moved to the batch header and eliminated from the line.
- Add the recommended local validations of operation/patrol/highway/crew and account number/cost centre.
- Increase the field sizes for hours and litres in the relevant screens.
- Allow addition and deletion of lines from a transaction in the MODIFY routine.
- Provide a facility for scanning all entries in a batch.
- Improve the documentation on the report generation system.

- Add vendor account files to replace names and addresses.

Satisfying any or all of these recommendations should have an impact on the system which is noticeable to users. We suspect that you will need to set your priorities among these recommendations based on the time and manpower involved in correcting them. Nonetheless, we believe that these problems are worthy of correction, if technically feasible. Other recommendations would improve the system, but not to the same extent.

3/ GUIDELINES FOR FUTURE SYSTEMS

In Section 2 we focussed on the specific strengths and weaknesses of the OMS and made recommendations for potential improvements. However, many of our findings relate not only to the OMS but also to form-filling interactive systems, in general. In this Chapter, we make more general comments relating to the design of form-filling interactive systems. These comments draw on both the review of the OMS and from the general literature available in the field.

3.1/ The General Principles Underlying Guidelines

Smith (1982) divides his guidelines into three major topics: data entry, data display and sequence control. For each of these topics, he provides a list of objectives which a designer should strive to accomplish. They fall into five general classes:

- efficiency,
- low memory load for user,
- consistency,
- compatibility,
- flexibility.

Efficiency refers to minimizing user effort. This involves minimizing input and control actions, minimizing errors, and minimizing user search times when looking at forms or screens.

Low memory load results when users are not required to remember commands or keep track of special cases in the data. Menus are used to avoid the need for user commands and well-designed forms can minimize the need for users to make decisions while inputting data.

Consistency primarily refers to predictable outcomes of user actions. The same key should be used to exit from all programs in a system. Either the user should manually tab from field to field at all times in form filling, or an auto skip function should always be used. (The former is typically recommended.) Finally, the actions caused by various system functions should be consistent with the user's perceptions of how the system should work. A strength of form-filling is that this match is

not difficult to achieve since the operations of filling in or correcting are not difficult and every user action is displayed on the screen.

Compatibility focuses on the relationship between data entry forms and screens. We have tended to consider this as a subtopic of consistency in the previous sections, although separate treatment of the way in which key strokes affect the system and the degree of conformity between screens and forms have been separately discussed. There is one other type of compatibility that Smith also mentions, compatibility of user needs with sequence control. The user should be allowed to do a task in a natural order -- not forced into rigid and unsatisfactory sequences of behaviour. This aspect is closely related to flexibility.

Flexibility refers to the degree of user control over data input, display and sequence control. For the data entry clerk, the most important flexibility will be the ability to correct errors during the data entry process without interrupting the input function. For management, flexibility may include the ability to design or modify forms through an easy-to-use form creation program.

We believe that the five design goals listed above are good general directions. Of course, a difficulty with guidelines is that there are tradeoffs. Optimizing one objective often degrades another. Fortunately, progress is being made in finding ways to optimize more of the objectives simultaneously. This is in part due to a greater appreciation of the kind of support users need to have. It also reflects the availability of more powerful computers at a reasonable cost and the growing willingness to provide a greater and better planned programming effort.

The form-filling, menu-selection approach seems to be one of the approaches which is most satisfactory for optimizing user satisfaction. However, such systems can vary a great deal in their satisfaction to the user based on some fairly specific design decisions. In the next section, we will attempt to cover some of the latest advice on a number of issues which are of direct concern. These issues are form design, screen design, the data entry process, menu design and operation, system

response time and display rates, feedback and system messages, and on-line help systems.

3.2/ More Specific Suggestions for Guidelines

3.2.1/ Form Design

This section and the next one are essentially a summary of the main points from Chapter 8 of "Human Factors in Office Automation" (Galitz, 1980). This chapter is highly readable and we urge anyone planning to work on forms or screens to take the time to read it.

The information on forms should be arranged according to a sensible combination of several factors.

Customary groupings -- Information is arranged in the order in which it is normally found. An address is given by street, city, province and postal code. Any deviation from this grouping will result in confusion.

Frequency of use -- Items used more frequently should be near the beginning of the form.

Function -- Items serving the same purpose are grouped together. That is, all the information used to identify a company, its name, address and phone number are given together.

Importance -- Important items are positioned in a prominent position, perhaps the first item in a new grouping.

There are many possible combinations of the above arrangements. A typical one is to organize the form into functional units, with items or customary groupings then organized by a combined consideration of importance and frequency. The overriding factor is that the data organization must make sense to the person providing or collecting the data on the

forms. Getting user comments and suggestions before and during trial use of a form is highly recommended.

After the organization of a form has been decided, the layout of the form must then be considered. Galitz offers the following advice:

- Provide sufficient writing space.
- Maximize visibility of fill-in fields.
- Provide intelligent and clearly stated captions.
- Clearly associate captions with fill-in fields.
- Incorporate coding aids.
- Use preprinted codes wherever possible.
- Provide clear design and clean reproduction by using perceptual groupings.
- Minimize irrelevant information. Use orderly and logical data sequence (top to bottom, left to right).
- Locate instructions at the top or beginning of the form.
- Develop for completion by handwriting, typewriter or both. (p. 146)

Failure to provide sufficient space so that information does not have to be crowded is a major source of legibility errors attributable to form layout.

The content of information to be coded also affects data entry errors. Research has shown that 47% of character entry errors involve I-1 or O-0. Thus, elimination of I and O from coding schemes will help users avoid substitution errors. In addition, it has been shown that 50% of the handwritten characters judged as illegible are Y N V 2 Q U and G. In fact Y N and V account for 25% of illegible characters. Thus, these letters should be avoided in hand printing except in cases where the alpha combinations are predictable and meaningful. That is, a hand written Y or N to indicate yes or no should not be requested, but the words YES or NO would not be confused.

In addition, Galitz presents evidence to show that long number groupings should be divided into subgroups of 3 or 4 digits in length to decrease error rates. As well, letter and number codes should not be interspaced.

Units of measurement should be included either in the label or as part of each data item; consistent formatting of data fields, such as giving all dollar amounts to two decimal places, is recommended; and frequently used data fields should be consistently formatted on different forms. That is, the date should not be dd/mm/yy on one occassion and mm/dd/yy on another. Phone numbers should be consistently given with or without hyphens between groupings.

Finally, forms should be designed so that users do not have to make decisions about changes or special cases while they enter data. According to Galitz, "The fewer rules and decisions involved in keying, the faster and more accurate data entry will be. Coding, omitting, changing or including data by special rules or transformations is probably the greatest single decrement to data entry speed." (p. 163).

3.2.2/ Screen Design

Rules about screen design depend on whether the screen is to be used with a form or for direct data entry. In the former case, the user's attention is primarily directed toward the form, not the screen; in direct entry the screen must provide all the information needed by the user. There are two major differences in form layout for the two cases. First, when data are entered directly, all captions must be stated in full on the screen; when forms are used, abbreviations or contractions of the captions shown on the form are given on the screen. Second, layout differs in the two situations. When forms are used, placement of entry fields on the screen should bear a direct relationship to their placement on the form; in direct entry, on the other hand, entries are aligned in columns, typically with one entry per row, as follows:

```
LAST NAME:          --------------------------------
FIRST NAME:         ------------------------
DATE (DDMMYY):      --------
```

Other than these two differences, the screens are designed in identical ways. In fact, design of a direct entry screen is identical to design of forms except that problems with legibility of handwriting do not occur

and much greater care must be taken to ask for information in the order which is most natural to the user.

When forms are used, it is important to place entry fields on screens into direct relationship with their placement on the forms to reduce user search time whenever the screen must be consulted. The order of items should be entirely consistent with the form. In addition, a user should be able to look for a physical grouping on the screen which is identical to that found on the form. This grouping rule does not apply to the captions, but only to the entry fields. Thus a form may have a long caption positioned above the field, but the screen might have short captions preceding the field. As an example, a form might have the following layout:

NAME (LAST, FIRST)

LOCATION	STARTING DATE	ENDING DATE	SUPPLY CODE
---	---	---	---
---	---	---	---

ORDER	HOURS	RATE	AMOUNT
---	---	---	---
---	---	---	---

On the screen the following might be observed:

Name: ---

Loc: --- S-Data: --- E-Date: --- S-Coded: ---

Ord: --- Hours: --- Rate: --- Amount: ---

While forms can be designed with captions either above or next to the fields, the latter arrangement should usually be used in screen design.

A column header is used for screens only when the fields are to be repeated two or more times on adjacent rows. For example:

Order:	Hours:	Rate:	Amount:
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----
-----	-----	-----	-----

Finally, a form may need to be broken into several screens. This is typically done in a way which keeps functional groupings of items together. If the form has been designed with more frequently used groupings ahead of those which are infrequently used groupings and with frequently used items within a group ahead of those which are often left blank, the user can often complete a form without having to display all of the screens or tabling through a lot of blank fields within a screen.

3.2.3/ The Data Entry Process

The information in this section is a composite of advice offered by Smith and Galitz. They do not always agree, in part because of the type of data entry they seem to have in mind. Galitz views form filling from the perspective of insurance form design. It has little redundancy in the information from line to line and entry is made using forms. The origin of Smith's guidelines are not as clear, but the following contrasts would seem to be appropriate:

Smith views the input clerk as doing a visual verification of each line before proceeding. This would probably occur in situations of direct entry without forms, but not when reading from forms. In addition, Smith is concerned with more general data entry, in which the same information is required in different forms. His general theme is "let the computer do the work".

- Users should not enter (unless for the purpose of data validation) any redundant information (e.g., both a name and vendor code).

- Appropriate defaults should be displayed in their appropriate fields on initial display of a screen.

- Data keyed in for one transaction should be retrieved when relevant to another transaction and displayed for user review.

- All calculations should be performed by the computer.

It is likely that Galitz would agree with all but the second recommendation above. However, he would be concerned that the user decisions about whether or not to use a default value would slow down keying and increase errors in data handling. Our observations indicate that errors do occur because of defaults, but that users like to have them. Our recommendations are as follows: If defaults are used, make them visible from the moment the screen is displayed. Do not use defaults for single column entries such as credit/debit because the decision time probably exceeds the keying time. Do use defaults, when reasonable, for longer field. If possible, set off the default entries on the screen by increasing the brightness of these items. It might be worthwhile to develop an easy to use "SET DEFAULTS" program which allows supervisors or even individual users to determine what defaults will be used for a given batch. Of course, a standard set of defaults will be available for users who do not wish to bother with the process.

Galitz and Smith generally agree on the data entry method. First, users should manually tab from entry to entry. even when there is a fixed field format. While auto skip to the next field would appear to be efficient, the fact that variable length fields are also found in most data sets means that partial use of auto skip violates the consistency requirement. Research clearly shows that consistency is more important in this case. All fields should be completed with manual tabbing. A cursor-rest position should be provided at the end of each line, as well. Users should be allowed to skip over blank fields at the end of a form without having to give multiple tabs. Finally, a user should be able to easily change any item in a transaction during the data entry process. The user should not be forced to terminate data entry or remember the error for later correction.

Special functions such as skipping fields and moving the cursor back to a previous screen should be available to the user by pressing a single key. This requires that the user's keyboard be equipped with function keys. If function keys are not available, users will have to learn escape sequences to accomplish the above activities, thus increasing the memory load and the keying effort. Or, as an even worse option, they can be trained to use both an insert and a command mode. Many studies have shown that the use of modes complicates the training process a great deal.

Assuming that function keys, or possibly escape sequences, will be developed, a minimum set of functions which should be available are given below.

Display next screen -- Avoids tabbing over unused fields by showing the next screen.

Display next form -- When a form has several screens, it is likely that no information needs to be placed on the last two or three. This key allows the user to skip over them.

Backspace -- Move cursor back one space without deleting the character.

Move cursor back one field -- Cursor positioned to the one field first character of the previous field. Contents of the field are now treated as a default which can be overwritten or left intact. (User can return to any previous position by this method).

Move cursor to first entry -- Cursor goes to first entry field of form; contents of form are treated as defaults and can be overwritten.

Move cursor to last entry -- Cursor goes to its most advanced position on the form, e.g., that position from which the backing up operations were initiated.

Cancel screen -- The contents of the screen are eliminated and the user can start again. The cursor is placed in the first entry.

Cancel form -- The user can eliminate the entire form. First page is again displayed, with original defaults.

Undo -- This key would allow the user to restore the state of affairs right before the last key press.

Smith would want separate TAB and ENTER keys as well. The TAB key would move the cursor to the next field without performing validation, while the ENTER key would result in a logically-related group of fields being validated (edited) together. Galitz would not want the use of the enter key to change from form to form as Smith seems to propose. This would be another form of inconsistency. However, the principle of using the TAB key to enter items within a line and the ENTER key to signal that the line is completed and ready for validation is commonly recommended.

The manner in which errors detected during the validation process are handled will depend on whether validation occurs field by field or line by line; whether type-ahead is supported or users are locked out of their keyboards until data entry is complete; and whether or not the user is working from a form. We cannot find good recommendations in the literature about error handling for these various cases, but will make an attempt at our own guidelines.

If direct data entry is being used, validation should occur when each field is entered, the user should not be allowed to type ahead, and a written error message should be given which gives as much information about the problem as possible.

If forms are used and type ahead is supported, the error handling should take into account the fact that users may be ahead of the action. One suggestion might be to signal a detected error by a bell and wait for user correction from the keyboard before continuing in the processing of the type-ahead buffer. The user could then be given the option of keeping or cancelling the buffer after the error has been corrected.

As computing power increases and more data are entered remotely, it is possible that type-ahead will become unnecessary due to improved response times, decreased use of forms, and decreased typing capability of those entering the data.

3.2.4/ Menu Design and Operation

In general, the same organizational considerations go into the design of menu items that go into the design of forms; natural sequencing, functional categories, frequency of use and importance should all be considered (see 3.1.2.1). In addition, it is important to word the menu choices in such a manner that the alternatives are clear to the user. Testing with users can lead to needed modifications.

Menus are not always successful. The choices per screen must be discrete and the logic of the hierarchy must be clear. So far, the OMS easily meets these requirements because very few options are actually used and the hierarchy has very few levels. Good menu design becomes more important as the size of the tree it is serving increases.

Consistency in the way that the user selects menu choices (e.g., press number of choice, then press the enter key) and the way in which he or she moves through the hierarchy is also important. For novice users, having prompts telling the user how to get back to the start or to the last screen is also necessary.

Experienced users find the need to work their way through lots of menu screens to reach a desired program very annoying, particularly when the display rates or retrieval times for each menu are slow. Many newer systems are supporting both menu selection and a method for skilled users to bypass menu selection. One method allows the user to bypass screens by a "choose-ahead" option. If a user knows the path to a target program (e.g., choose 3 on this screen, followed by 5 on the next and 4 on the next), the intermediate screens can be bypassed by typing all codes successively (e.g., for the above example, 3, 5, 4 would be typed on the same line before the enter key was pressed.) Another system for experienced users involves giving alphanumeric names to key screens so that knowledgeable users can directly call the target by name. We would

recommend that one of these systems be incorporated into future menu-selection programs.

3.2.5/ Response Time and Display Rates

User satisfaction with display rates depends on the experience of the user, the task, and the type of mental activity involved. A recent summary of the literature by Schneiderman (1982) includes the following:

- Typing and cursor motion commands should generate results in 0.1 seconds.
- Frequent simple commands should take less than a second.
- Other commands can take longer, but users will attempt to structure their work to avoid long delays.

With respect to form-filling, it is preferable if the system can keep up with the user. It is frustrating to users when this is not the case and causes problems for field validation and error handling. Slowing users down also results in decreased throughput.

It should be noted, however, that the speed required to keep up with the user varies from application to application. Two situations, in particular, can be expected to place fewer demands on the system. When data are entered directly, without forms, users are not likely to attempt to type ahead. Also, when the users as a class, tend to have poor typing skills, a moderately responsive system would be adequate.

3.2.6/ Feedback and Error Messages

Dean (1982) emphasizes the need for planning system messages from the earliest stages of the design process.

> "I have noticed how programmers usually try to create good messages. After their programs are practically finished, they get someone to edit and reword the messages that have sprung up like weeds during program or system development. Thus, they try to ensure that essentially

unplanned messages are concise, grammatical, consistent, and understandable. But these qualities do not guarantee that the messages are also relevant, specific, timely, and helpful. I have concluded that you cannot edit and rework the second set of qualities into messages that already exist." (p. 425).

Users should always be given an indication that a valid command has been received and processed. The feedback might be simple and natural, as when the cursor tab key results in movement of the cursor on the screen. Or a special message may be required. Thus, if a user requests that a file be copied, a message may be required. A message such as "COPYING" should be displayed, followed by "COPY COMPLETE" when the task is completed. These messages should be displayed in a standard place on the screen where the user can easily view them.

There are limited conditions in which an auditory signal may be used as a feedback sign. First, experienced users who are entering data from forms need an auditory cue when errors are detected because their attention is not directed toward the screen. In general, when the bell is used to signal an error, it should not be used for other purposes during data entry. It is important not to provide the user with "false alarms" or they will begin to ignore the bell. One other situation in which a bell may be useful is to signal that the computer has completed a task which routinely takes a long time to complete. By this, we mean a time which is so long that the user will do other work while the job is being processed. An auditory signal is appreciated by users under these conditions because they do not have to continually check the screen to determine when the work is done.

Schneiderman (1982) considers the wording of error messages to be particularly important for inexperienced users. Based on his own research, he provides the following guidelines:

- Avoid negative words such a "ILLEGAL", "INVALID", "ERROR", or "INCORRECT".

- Use words from the user's own problem domain to describe a problem. For example, "Pay rates range from $5 to $30."

- Place the user in control of the situation and provide enough information to allow the user to take action. Corrections should be made by overstrike so that the user can see the error.

- Have a neat, consistent, and comprehensible format. Error messages should always appear in the same part of the screen, perhaps the bottom two lines.

- Before error messages are implemented finally on a system, users should be asked to review the messages. They should indicate that messages clearly identify both the error and how to correct the error.

The third guideline above requires some adaptation to the form-filling situation in which we are trying to give the user a great deal of feedback. There we recommend display of the form on the screen and the use of cursor positioning to mark where the user is working. Cursor position will also be a strong and natural cue to the location of the error on the form when type ahead is allowed. We recommend that the cursor be automatically positioned to the start of the field needing correction when an error is detected. The value which the user had previously typed in the field should remain displayed while an appropriate error message is given in its normal position on the screen. Many times, a simple comparison of what was typed and what is written on the form will lead to the detection of a simple typing error which the user can overwrite.

Finally, while not dealt with in the guidelines, we would like to comment on how long an error message displayed on the screen should be. The OMS tends to display messages for a given time period. Because the user often prefers to look at what has been typed and at the form first, this time can be too short. A simple but effective rule which avoids this problem is to cancel the message when the user either successfully corrects the error, or presses a special function key such as the "help" key.

3.2.7/ On-line Help Systems

On-line help systems are becoming increasingly popular. First, they make it possible for the user to get help without leaving the terminal to find

the never-present manual. In addition, if the on-line manual is thorough, it can replace the printed manual and be updated at low cost. Finally, in theory, an on-line system can provide the user with information appropriate for the current task.

Unfortunately, developing a system on-line does not bypass problems of writing in a manner that is comprehensible to the user, nor does it solve the user's problem of locating the needed information. Careful planning must go into an on-line help system if it is to be effective.

The kind of information the user can find by using help commands must be carefully defined so that the user will know the types of information available. For example, help might be available only for explaining the function keys and command sequences available (such as how to exit a given program). Or, it might offer more detailed instructions for filling in each form. It could allow users to see a list of current defaults for a table or examples of how to set up reports. Whatever decision is made, the boundaries of the help system needs to be clearly stated so that users will not waste time looking for a type of information which is not there.

The user should be able to get help by pressing a single function key, usually to the extreme right or left of the function key row. The appropriate help screen should then be displayed. The contents of this screen will differ, depending on the context in which the help key is pressed. On the other hand, the format of the screen should be consistent with other help screens to make scanning less of a problem for the users. After the user has used the help facility, a single key stroke should redisplay the page from which help was requested.

The major function of the help system will probably be reference as opposed to training. Recent studies have all shown that new users learn more poorly from on-line aid than from printed manuals. Thus, having an experienced user show a new user how to use the system, a method generally superior to learning from manuals, should also be better than on-line help for this purpose.

While users should be able to produce a printed copy of the manual, it is recommended that the original be maintained as a computer file, easy to update and produce on local printers as needed by the users. A facility for keeping a computer log of user problems and their solutions as they arise might also be very helpful, particularly if the person who maintains the file keeps it well organized. Thus, a supervisor could dump the part of the log dealing with "transmission problems" or "problems encountered with the accounts receivable form" which would provide a history of problems and solutions.

3.3/ The Need for User-Testing

3.3.1/ Review of Existing Systems

The review of the OMS system has indicated that a structured examination of an interactive system "in operation" can yield many benefits. It can provide information on:

- The strengths of the system,
- Areas of real or potential problems (including an estimation of the comparative importance of these),
- Possible solutions to these problems.

In addition to providing information on how the system might be improved, it also provides the user an opportunity to comment on the system's performance and participate in the improvement process. Even if the user has been heavily involved in the systems development process, it is only after a period of actual use of the system that most users can provide informed input on the user-system interface.

This suggests that a review should be carried out at some stage in the life cycle of every interactive system.

Ideally, this should be done six to 12 months after the implementation of the system. At that time, the user will have gained sufficient familiarity with the system to provide informed input, yet it is early enough in the system's life cycle that major benefits can be obtained by change. Indeed, this type of review might form part of a more general post-implementation audit.

3.3.2/ Review During the Design Process

The guidelines reviewed in this report could form part of the systems development process. A number of approaches are possible.

- At a mininum, systems designers should be made aware of the guidelines and the body of literature.

- The general guidelines given could be expanded and structured into a "standard" which form part of the general systems development standards within the Ministry. This standard would be provided to project teams as a mandatory requirement in the development of an interactive system.

- Specific responsibility for users system dialogue could be included as a specific project responsibility. Thus, an individual within the project team would have responsibility for:
  - disseminating the standards,
  - providing advice and guidance to the systems designers,
  - performing a quality control function throughout the design process.

3.3.3/ The Use of "Prototype" Systems

In many large scale interactive system projects, use is made of the "prototyping" concept. That is, once the initial systems design has been established, the study team builds a preliminary model of the system. This is a particularly effective way of permitting both designers and potential users to see "in action" the system as it might operate. This technique is particularly appropriate for examination of the user-system dialogue. It combines aspects of the design process review and of the post-implementation review. By involving specialists, systems designers and users in a structured review of the prototype system, many lessons can be learned and improvements made to the system prior to its general release.

This type of approach may also be appropriate in situations where a "pilot" implementation is carried out; that is, where the system is implemented in only part of its intended user community. This limited scale field trial can also be used for a review of the user-system

interface. However, in this case, it will be more difficult to make major changes in the systems design since the total system will normally have been developed. In contrast, the prototyping approach assumes that such changes might be made and the level of effort to make changes will be significantly less, as will the amount of "wasted" work that might be replaced.

3.4/ Literature Identifying Guidelines

During this study, many papers, articles, and books related to the topic of user-system interfaces were reviewed. They are all listed under References. However, four of these references merit highlighting in the body of this report. These are references which should be available to designers developing form-filling systems. Their form and clarity will make the short time required for reading them time well spent. The key references are described in subsequent sections.

3.4.1/ W.O. Galitz, "Handbook of Screen Format Design", Q.E.D. Information Sciences, Inc., 1981

Galitz is one of the most prolific authors in this field. In this book, his objective is "to provide materials to assist a designer in developing an effective screen interface between a program and its intended users". Specifically, it sets out to enable the reader to:

- describe the considerations that must be applied to the screen design process,
- describe a series of design rules that can be applied to several categories of screens,
- perform the design steps necessary to develop and layout effective screens.

The book includes:

- a review of two major concepts, design consistency and design trade-offs;

- considerations integral to the design process (human factors, software, hardware, and applications);

- specific types of screen (e.g., data collection, data entry, enquiry, interactive, and menu screens);

- colour display;

- source document design;

- the design process to set out a typical data entry screen.

The book is generally well-written, although somewhat verbose, and contains many examples of different aspects of screen design. While it is not always easy to relate his recommendations to specific material in the research literature, this book provides the best generally available review of good design techniques that can be used by the systems designer. Of particular note, is the inclusion of a chapter on source document design. This is an area which is seldom given sufficient attention during the design phase.

3.4.2/ W.O. Galitz, "Human Factors in Office Automation", Life Office Management Association, 1980.

This book is somewhat of a contrast to the other Galitz handbook described here. Despite its title, its content relates as much to computer systems as it does to office automation. It includes much background material, including appropriate references covering a wide variety of the issues related to "operator-machine" interfaces. Some of the material is a re-presentation of that included in the handbook, but the majority of it is complementary information. Its importance to the systems designer, is that it covers a wide range of topics in a very concise and readable style. These include:

- a "layman's language" discussion on the human factors related to the use of computer systems;

- a review of the various components of the "office environment":
  - lighting
  - acoustics
  - workstations
  - office layouts;

- systems design guidelines;

- interface guidelines (including non-keyboard interfaces);

- information transfer guidelines;

- documentation guidelines;

- office environment guidelines.

This is an excellent document for the systems designer who has little exposure to human factors and "ergonomics" issues. While it contains many practical examples, it is less directly applicable to the screen design process than the handbook.

3.4.3/ S.L. Smith, "User-System Interface Design for Computer-based Information System", A Report Prepared for the United States Air Force, April 1982.

This report was prepared by the Mitre Corporation, Bedford, Massachusetts, for the Air Force System Command, United States Air Force. It is approved for public release with unlimited distribution. Mitre Corporation is a vendor of computer communications equipment and consulting services, operating largely in the public and defense sectors. Their report has four main sections:

- a background review of the present "state-of-the-art" in user-system interface (USI),
- definition of USI requirements,
- USI design guidelines (including a brief literature review),

- USI documentation.

In addition, there are five appendices. These include a checklist of USI functional capabilities (designed for use on a feature-by-feature basis by systems designers), and design guidelines for data entry functions, data display functions, and sequence control functions.

Smith suggests that USI guidelines must cover six areas:
- data entry,
- data display,
- sequence control,
- user guidance,
- data transfer,
- data protection.

Work is continuing at Mitre Corporation in all six areas.

This report is an excellent detailed review of the present state-of-the-art in user-systems interface design. Of particular interest to this project were the detailed design guidelines included in the appendices. Covering some 110 pages, these contained comments on virtually every situation likely to be encountered in designing the user-system interface, with recommendations for the most appropriate design approach. For each recommendation, back-up references justifying the approach are provided. The main problem the reader will encounter in this area is the sheer volume of the recommendations. For these to be really effective to most designers, some form of analysis and abstraction is likely to be necessary. It should also be noted that a number of the recommendations have a direct relationship to the military systems nature of the Mitre contract. Thus, it includes consideration of such non-traditional systems as tactical command situations.

3.4.4/ M. Dean, "How a Computer Should Talk to People", I.B.M. Systems Journal, volume 21, No. 4, pp. 424-453, 1982.

Dean's background is that of a programmer, systems engineer, and technical writer. In his paper, he sets out to provide "common sense insights into the philosophy of creating good computer messages". This reference

is particularly appropriate for systems using a command language, rather than form-filling systems. Bearing this in mind, Dean suggests that systems messages should have two sets of qualities.

- They should be "concise, grammatical, consistent, and understandable".
- They should also be "relevant, specific, timely, and helpful".

He suggests that to create messages with both sets of qualities, certain things have to be done. His essay considers them under the following headings:

- Set human goals for messages.
- Apply psychology in writing messages.
- Write messages that accommodate intended users and their situations.
- Playact to evaluate the messages for usability before coding.
- Edit the messages for appropriate language.
- Design the computer program or system to produce messages.
- Test the messages along with the running code.

He then provides a detailed discussion under each topic, including examples of bad and good system messages and discussion of alternatives available in each category.

3.5/ Trends in Workstations

This project has focussed on the issues related to the operator-system interface in what might be described as a "traditional" terminal environment; that is, the use of a workstation, consisting of a keyboard and visual display screen, set-up permanently in an office environment, connected to a larger remote computer. While this type of situation is likely to remain a significant part of the total environment, there are significant new trends developing. It seems appropriate to conclude this report with a review of some of the most significant trends.

3.5.1/ Distribution of Intelligence

There is a steady move towards "distributed data processing". Indeed, the OMS reviewed in this project uses equipment specifically designed for a distributed or dispersed environment. This trend is likely to continue. The availability of very powerful micro-processor-based systems

(usually based on a 16 bit architecture), along with the much lower cost of mass storage devices will accelerate this process. Thus, the systems designer will have to consider two levels of user system interface -- the user-local system and the user-local-central system interfaces.

3.5.2/ The Move Towards "Executive" or "Professional" Terminals

Until recently, computer terminals were almost exclusively found on the desks of various levels of clerical employees. This is no longer the case. Increasingly we are finding workstations on the desks of professional and executive workers, the so-called "knowledge workers". These terminals may be used to access and inquire about corporate databases, to use various decision support tools (such as, spread sheet modelling), or to provide access to various office automation facilities (e.g., electronic mail). This has caused the appearance in the marketplace of a wide variety of terminal designs. These terminals may vary in physical appearance, cost, and facilities offered. Thus, the systems designer is now faced with the situation that all users will not necessarily be accessing the system through the same type of terminal. Therefore, the designer cannot assume that each user will have the same set of terminal facilities.

3.5.3/ The Use of Multi-Function Workstations

As suggested above, as terminals become more widely dispersed through the office, they may be used for more than one purpose. Not only may the same workstation be used to access several different computer applications on the same host computer, it may also be required to access facilities from different host computers. This raises many issues for the designer of the user system interface. For example, can the interfaces between systems and between host computers be made sufficiently similar to avoid unnecessarily confusing the user. Further, the systems designer may have to consider the "pass through" concept. That is, the user may wish to access computer system "B" by having the connection made through system "A".

3.5.4/ Changes in the Basic Design of the Workstation

We are also seeing major changes in the ways that workstations are being designed. This is happening in two ways:

- Multi-media workstations. Some of the workstations coming on the market today, combine multi-media communications. For example, the Northern Telecom "Displayphone" provides the user with both a voice and a data communications facility.

- Other, recently announced, systems have changed the way in which users interact with the system. Traditionally, the alpha-numeric keyboard, with additional function keys, has been the main tool for the user to communicate with the system. The only exception to this has been the use, mainly by IBM, of the "light pen" to select certain items on the screen. However, the light pen has not found widespread acceptance. Now, new approaches are being considered. Two recent ideas are:

  - The touch sensitive screen. Some terminal equipment now available, allows the user to select items on a menu by simply touching the screen beside the item to be selected.

  - Non-logical cursor controls. Traditionally, the user has controlled the cursor position on the screen by various buttons on the keyboard (e.g., the left-right, up-down control pad). Recently announced workstations such as the Xerox "Star" and the Apple "Lisa" make use of the "mouse" and "icon" concepts. These facilities permit the user to interact with the system using "visual" rather than "logical" processes. In some ways, this could be compared to many of the techniques used by the video game industry.

3.5.5/ The Move Towards Portable Terminals

There are many terminals now in the marketplace which permit the user to access computer systems through direct-dial facilities. These terminals include such wide variations as:

- The portable memory printing terminal. This type of terminal, typically with an internal memory of some 3000 to 4000 characters, provides a full typewriter keyboard and a printing facility (normally using heat sensitive paper) which permits the user a high degree of typing capability.

- The compact portable. Often little bigger than a pocket calculator, these systems usually include a small keyboard and a one-line video display. Frequently, a small printer can be added to this device.

- A portable micro-computer. In the latest wave of micro-computer releases, there have appeared several products which include powerful micro-computing capabilities with a significant amount of portability. Examples of this include the "Osborne I" and the "Hyperion".

3.5.6/ The Increased Focus on "Ergonomics"

Ergonomics, the science of person-machine interface design, has received a much greater focus in North America in the early 1980's. Athough frequently the VDT itself is seen as a major user-system interface issue, we must also consider the total environment in which the user is working. Thus, such physical considerations as desk and chair design, keyboard and screen design, and light and heat issues are now receiving significant attention. In addition, there are the psychological issues. Several recent studies have indicated that there are major psychological problems involved with tHe use of visual display terminals in the office. Such factors as higher stress and apparent increased susceptibility to certain types of illness have been identified. Most organizations, when introducing VDT's, must take much greater consideration of these issues than in the past.

REFERENCES

[1] Anderson, R., "Value Orientations of Computer Science Students," Communications of the AMC, 1978, 21, 3.

[2] Bernard, P.J., Hammond, N.V. J., Long, J.B. & Clark, I.A., "Consistency and Compatibility in Human-Computer Dialogue," International Journal of Man-Machine Interface, 1981, 15, 87-134.

[3] Benbasat, I., Dexter, A.S., and Masules, P.S., "An Experimental Study of the Human/Computer Interface," Communications of the ACM, 1981, 24, 752-762.

[4] Carey, T., "Software Interface Requirements for Office Communications Systems," Report to the Department of Communications, Ottawa, 1982.

[6] Dean, M., "How a Computer Should Talk to People," I.B.M. Systems Journal, 1982, 21(4), 424-453.

[7] Denzin, K.D., "Strategies for Analysis of the Task," In Hartley, J. (ed), Strategies for Programmed Instruction: An Educational Technology, London, Butters-worth, 1972.

[8] Ericsson, K., and Simon, H.A., "Verbal Reports as Data," Psychological Review, 1980, 87, 215-251.

[9] Flanagan, J.C., "The Critical Incident Technique," Psychological Bulletin, 1954, 51, 327-350.

[10] Galitz, W.O., "Handbook of Screen Format Design," Q.E.D. Information Sciences, Inc., 1981.

[11] Galitz, W.O., "Human Factors in Office Automation," Life Office Management Association, 1981.

[12] Moran, T.P., "An Applied Psychology of the User," ACM Computing Surveys, 1981, 13(1), 1-11.

[13] Norman, D., "Steps Toward a Cognitive Engineering: Design Rules Based on Analysis of Human Error," Proceedings of Human Factors in Computer Systems, Gaithersburg, Maryland, 1982, 378-382.

[14] Ramsey, H.R., and Atwood, M.E., "Human Factors in Computer Systems: A Review of the Literature," Technical report SAI-79-111-DEN, Science Applications Incorporated, Englewood, Colorado, under contract for the Engineering Psychology Programs, Office of Naval Research, September, 1979.

[15] Schneiderman, B., "The Future of Interactive Systems and the Emergence of Direct Manipulation," Keynote address at the NYU Symposium on user interfaces, 1982.

[16] Schorer, P., "Structure the Use: Notes on a Method for Designing Computing System Environments," Computer, December 1981, 77-86.

[17] Smith, S.L., "User-System Interface Design for Computer-Based Information Systems," The Mitre Corporation, a report prepared for the United States Air Force, April 1982.

[18] Tombaugh, J.W., "Methods for Applied Research for Computer Applications," Behaviour Research Methods and Instrumentation, in press.

[19] Tombaugh, J.W., and Dillon, R.F., "Recommendations Concerning Human Factors Guidelines for Designing Office System Manuals," Report prepared for the Behavioural Research and Evaluation Group, Department of Communications, Ottawa, March 1982, in press.

APPENDICES

APPENDIX A/ METHOD DOCUMENTS

Supervisor (and Operator) Interview Protocol for Task I Site Visits

I/ DESCRIBE PERSONAL USE

A/ Applications (Identify your three major applications)

1/ Types of application, e.g., cost centre budget transaction, maintenance daily work report, etc.
2/ How often is the system used for each application?
3/ Extent of personal involvement in the application.
4/ Reason for involvement.
5/ What was your involvement before the interactive system was installed? If changed, why? How?
6/ How did you keep track of this information before?
7/ Is the original system still in place -- maybe in a scaled-down version?
8/ Do the old and new systems keep track of exactly the same information? What has been added or deleted?
- For add: How often and how do you use the new information for decision-making.
- For delete: Is there a substitue? How was the information used before for decision-making? What do you do now?

9/ (a, b) In what ways was (is) it easier to deal with this information before (after) the interactive system?
10/ What changes in the system would make your job easier at this point.

B/ Tasks Done on the System (Identify your three major tasks)

1/ Types of tasks done -- sign-on, start-up, data entry, data verification, inquiry, report requests, end of day run, etc.
2/ How often do you do it? Who usually does it?
3/ Why do you do it -- as opposed to someone else?

4/ If you run into a problem trying to do something with the system, what do you do about it? Help commands? Manuals?
5/ When you do the task, how do you check that it has been done right?
6/ What was your involvement before the interactive system? If changed, why? How?
7/ How was this done before? (if appropriate)
8/ Do you have any back-up system for checking task accuracy? (if appropriate)
9/ In what ways was (is) it easier to do this before (after) the interactive system was installed?

C/ General Problems

What difficulties do you find with:
1/ system security and control,
2/ data entry,
3/ error correction -- data validation,
4/ display of data on the screen,
5/ dialogue/menus,
6/ inquiry,
7/ update of data bases,
8/ getting reports.

D/ Prior Computer Experience

- used computers before,
- taken courses on computers,
- written computer programs,
- relevant work experience,
- relevant educational background.

II/ OPERATORS

A/ Quantity

1/ What categories of employee are in your office?
2/ How many users in your office?

3/ How much of their day is spent on the system? Preparing data for the system?
4/ Were any of these people hired specially to do this work -- did you get a staff increase to do this?

B/ Quality

1/ What non-computer activities, if any, have been displaced (added) by the interactive system?
2/ What effect does this have on the work of your office? Pros? Cons?
3/ Is your data more reliable now? Why (not)?
4/ Are you finding more errors in your reports than you did before? Why (not)? What type?
5/ Has the interactive system had any effect on your staff turnover? Why?
6/ Has the interactive system had any impact on the job satisfaction of people who are now operators? Why?

III/ MANAGERIAL USE

A/ Usage

1/ Who uses it/receives information from it?
2/ What do they use the system to do? e.g., reports
3/ What reports do they request?

B/ What complaints or compliments have your heard about the system? From whom?

C/ Do you think it has made the manager's job easier? How? Why?

IV/ OPINIONS

1/ Has the interactive system been for the better? Do you prefer it to the previous system? Why?
2/ What did you like better about the previous system?
3/ What complaints do you have about the current system?
4/ Has the computer produced more advantages or disadvantages? Why?
5. Are you finding yourself too dependent on the computer? Why?

Task I -- Summary of Activities

1/ Familiarize yourself with the system. Walk-through an input/output task including corrections, etc.

2/ Supervisor interview -- following outline.

3/ User interviews -- 2 or 3 operators.

4/ Complete your section of the Smith checklist:
- Note guidelines/features met and/or not satisfied.
- For guidelines not satisfied, identify the problem and probable action for correction.

5/ Compile your supervisor and user interview reports following the interview outline. Identify key issues and usage problems in a summary. Submit interview reports.

6/ Write up your section of the guidelines in brief form for an interim report. Add general comments and submit. Note major issues outside but relevant to the interactive system, e.g., reliability of input information, preparation of information for input, etc.

7/ Identify questions for the manager and user surveys.

Observation Form: Project 7249 Dialogue

Site: Observer:

Type of Batch (Transaction type):

Timing: Start: : End : Elapsed: :

OBSERVATION

ITEM	COUNT	HASH MARKS
• Speed		
- # of Transactions		
- # of Lines		
- Est. # of keystrokes needed		
• Errors		
- # Backspace sequences		
- # Bell sequences		
- # Bells per sequence: 1		
2		
3		
- Est. strokes lost from type-ahead: 1		
2		
3		
• Error Type		
- # Automatic		
- # Stop and search screen		
- # Stop and search documents		

QUESTIONS -- RE: OBSERVATION

- Was this a typical batch? Why?
- Were you keying at your normal speed? How different?
- Did you make more of fewer errors than usual in this batch? How different?
- How did having me watch you affect your work?

ERROR CORRECTION

- Corrections during data entry -- would you correct an error and how?
  (Record actions for each sequence)
  - current screen
  - earlier screen for the same line item
  - earlier item in the same transaction
  - transaction header
  - earlier transaction

- Tell me what you do next with this batch?
  - procedures
  - reasons

- If you have keyed something during data entry that you know is wrong, do you prefer to change it during data entry or by using the MODIFY routine?

- Aside from finding changes you need to make during data entry, batch proof review and start-of-day messages; how else do you find things you need to change in your batch?

- Show me how you use modify to:
  (Observe and record the movements through the menu when performing various modifications.)
  - Add two lines to a transaction -- one in the middle and one at the end.
  - Modify two lines.
  - Delete two lines.
  - Add a transaction at the beginning, middle and end of the file.

SEMI-STRUCTURED OBSERVATION

Reports

- Routine reports.
  - What are they -- daily and monthly?

- Action needed to request them.
- Set-up needed for daily action and modification.
- Watch the set-up, observe menu movements and check against guidelines.

- Special requests.
  - Identify common requests.
  - Identify action needed to implement request.
  - For each request, set-up and modification: what is being done, observe menu movement, compare to guidelines.
  - Get a list of a report request files for the day.
  - Find out what happens to report numbers at the start of new year.

INDEPENDENT WORK

- Check out guidelines, using a different portion of the guidelines in each of the two sites. Make sure that each of the three observers uses a different portion of the guidelines in each office.

- Check out the match between forms and screens.

- Examine data repetition in data entry, especially for typical batches.

- Examine organization
  - user level and function,
  - office characteristics per criteria in relation to operations.

APPENDIX B/ DESCRIPTION OF THE OMS SYSTEM

The System

The terms of reference for this study gave an accurate overview of the OMS:

> The OMS is a distributed computer system with Datapoint mini-computers in each of the Ministry's five regional and 13 district offices, and in Head Office. Data are collected from the regions and districts and transmitted by autodial to the central processing site in Downsview. This is done at the end of each day and one jobstream is run every night to process all the data which were collected from the regions and districts. During this processing, central files and data bases are updated and reports are generated. These reports, along with any messages and audit trails which were produced during the run, are then sent back by autodial to the offices to which they apply. Reports can contain accounting information (expenditures, budgets by cost centres, project cost, etc.), management information (maintenance management operation, etc.) or other information as required.
>
> The interactive part of the OMS is limited to its Remote Subsystem; the part of the system implemented on intelligent terminals in the Regional and District Offices and the Financial Planning and Administration Branch, Downsview. The main interactive function is data entry.

The command language for the interactive part of the OMS consists entirely of menu selection and form filling. A daily work cycle includes a 'start of day' procedure, data entry throughout the day of a variety of types of information under control of the local computer, and an 'end of day' procedure which clears the way for overnight transmission of the balanced batches to the central processing site at Head Office.

The user typically inputs a batch of transactions from a form. Each batch has a batch header which indicates the number of transactions and hash totals which have been computed with a printing calculator. The display screen for each transaction consists of labels followed by a field indicator in which the required information is to be coded.

The user has some ability to correct errors during data entry. After data entry, information can be corrected or changed using the "modify" routine available on the system.

Corrected batches are then prepared for transmission.

The Users

The method of integrating the OMS into the office varied considerably from one office to another. In some cases, one or two operators handled most of the input. Typically, these operators were highly experienced with the system and permanently assigned to their task; although one office rotated new people into the position every year or two.

In contrast to the offices where most of the work was done by a single operator, other offices used a system in which work was distributed. Each clerk entered data from the batches which they prepared. Usually in these offices, no one operator would use the system more than two hours per day. In addition, one or more experienced operators would be used as resource people for training, answering questions and doing the lower volume system control operations.

Typing speed varied from highs of around 180 characters per minute for the excellent typists to a low of about 60 characters per minute for a very inexperienced operator. The higher rate is probably a limit imposed by system response times, since faster users always type ahead of the system and are forced to wait for field validation at various points in the input process.

Despite the wide variation in typing speed on the system, all of the users we observed shared certain highly desirable characteristics. Their accuracy, understanding of the forms and ability to use the computer system were always outstanding. In addition, while most of them would like to see changes, they conveyed an attitude of general satisfaction. With the exception of anxieties about transmission problems, most of the suggestions for change are suggestions for improving throughput or decreasing user frustrations in a generally good system.

APPENDIX C/ SYSTEM PROBLEMS

Inconsistency

Inconsistencies are presented in three groups: system, form, and report inconsistencies. System inconsistencies refer to problems encountered in menus, default options, and exit procedures. Form inconsistencies refer to differences in location of information on a source document as compared to the screen. Report inconsistency refers to problems in requesting reports and in the kinds of information available from them. All of the inconsistencies we identify are summarized in Table 1.

System Inconsistencies

There are a number of inconsistencies at the system level. But, few are significant given the nature of OMS. These relate to menus, default options, data entry, and exit procedures. The major problems identified with menus are:

- Options on the menu are not always in production.
- Order of options listed in the menu do not reflect either frequency of use or other logical grouping.
- There is no standard entry window (position and size) for entry of menu selections.
- In some cases, main menu applications have no intermediate option screen.
- The terminology describing menu choices is inconsistent. For example, "entry or selection", "selection by number", and "select one of the above functions".
- There are varying data definition requirements for field entry, for example, "03" and "3" as a menu choice.

Despite these problems with menus, we believe there is really little long-term impact of these problems upon users of this system. These things may slow down learning time for a new user, but should have no impact upon experienced users of the system.

Procedures for exiting from routines, on the other hand, remain an annoyance to experienced users. Some routines are exited using a "99", some using a "5", and some using an "*". Other means of exit may also occur. It is difficult for operators to remember which method of exit is the one to use in each routine. Thus, they will not always know how to escape when they encounter a problem in a routine. In many instances, such as the "destination-location" screen on a report request, there is no explanation of how to exit. Many operators report being trapped in fields without knowing how to enter new data or how to exit. This can have a real impact on operator efficiency and even more so on operator satisfaction with the system.

Default inconsistencies have an impact on operator errors, as well as posing an annoyance in system operation. We have identified several default inconsistencies including:

- Accounts receivable defaults to credit, while all other routines default dollar amounts to debit.

- Invalid data fields are treated differently. Sometimes the system clears the information in that field, while other times the operator must delete it.

- The display of error messages sometimes clears in a few seconds and other times remains until action is taken.

- In some instances, fields which are to be bypassed require a "0" entry in order to bypass them. This happens, for example, in the highway/-crew/patrol number fields on maintenance management reports.

This section has identified many system inconsistencies. The most significant problem is the lack of a consistent method of exit from routines.

Form Inconsistencies

Most inconsistencies in this section are small differences in the location of information on the source document and on the screen. In general, the relationship between source documents and screens is excellent. But for almost every input document, there is some small difference between its layout and that appearing on the screen. In some instances, such as journal entries, it is because the screen is used to enter a variety of different but related input forms. In other instances, the reason for the inconsistency is less clear.

Before discussing each particular input form, we would like to relay a few observations:

- There is lack of common "style" to screen layouts.
- Data entry from forms sometimes moves forward by pressing the "enter" key; but other times the cursor seems to move forward without any direct action -- usually skipping over fields.
- Requests for "operation code" information in the maintenance system are inconsistent with the way work is done. The requests ask for information relative to one operation. But, most trips requiring work are multipurpose in nature. Thus, artificial breakdowns are usually made for the purpose of data entry. The form, then, becomes a convenient fiction.

Now, let us look at the correspondence between source documents and screens for a number of common data entry forms.

Batch Contol Slip

- "Location" on the form is not entered on the screen.
- "Total dollar amount net" on the form is not entered on the screen.
- "Local office reference number" on the form is not entered on the screen.

- "Total dollar amount hash" but everyone we talked to believes this should be the total dollar amount net. People find the hash total a not very helpful way to check on the batch totals.

The bi-weekly period maintenance posting document has a number of problems. These include:

- "Home location" on the form is "district" on the screen.

- "Structure number" appears on the screen but it is not on the source document at all.

- "Sub" appears on the entry form, but no one knows its purpose.

- "Acc Qty" on the screen corresponds to "Total Accom. Units" on the form. As well, this quantity is entered before total hours worked on the screen, but occurs after total hours worked on the form.

- "Total hours worked" on the form allows for numbers up to 999.9, but the screen allows entries only up to 99.9. (This also applies to garage work orders.)

- "Time worked" is next to "rate" on the form. The proximity of a number with one digit after the decimal point to a number with two digits after a decimal point poses a problem for some operators. The same holds true further along on the form for "quantity" and "unit cost".

- The form allows quantities up to 99 999.9 litres of fuel; while the screen only allows quantities up to 9999.9 to be entered. (This also applies to equipment costs.)

Accounts receivable has the contract number out of sequence. This poses a real problem for many people in locating it. Less of a problem, but also out of sequence are the dollar amount of the transaction and the credit/debit indicator. In accounts payable, the credit/debit indicator is also out of sequence.

Branch transfers also have a few sequence problems. These include the transaction type, the credit/debit indicator, and the batch number. These are also out of sequence on municipal subsidies.

When looking at invoices, you read down the page and first encounter name, then date, then local office reference number. When they are entered on the screen, the sequence of entry is local office reference number, date, and then name. This is virtually the reverse sequence of the form.

In cost centre budget transactions, the effective date is out of sequence. As well, "location centre" on the form is "location code" on the screen.

In funding work order transactions, the sequence of entry of the special project code and the work order number are reversed. Also, "acc type" and "assoc no." appear on the screen, but not on the form.

In the form used for district/crew table update transactions, the effective date is out of sequence. In addition, the crew description appears on the screen but not on the form. "Stock account" also appears on the screen but is not on the form and is not yet usable.

We also observe that the effective date is out of sequence on several forms, including equipment item transactions; location, patrol, highway table up-date transactions; and the pay rates table up-date transaction.

Furthermore, we note the location code is out of sequence on the end of period transaction.

In addition to these standard parts of the OMS, we have some observations on the garage work order.

- There is no credit column on the form. There is just red writing on the voucher jacket to indicate a credit. Credit must be entered as a negative number in the program.

- Each form is verified for a net dollar amount, but is balanced for hash total only. If the batch doesn't balance, you must note or copy down totals because you will never get to see them again.

- Background detail on vehicles is often recorded on scraps of paper attached to the main source document.

Once again, we have noted many detailed problems. Of these, we think the most serious is the lack of a sufficient number of digits for hours and litres of fuel on the maintenance management screen.

Report Inconsistencies

Report inconsistencies deal with problems in using the report request routine, problems of report dating, and the problem of information which is entered on data entry but not available in reports. The problems people encouter depends largely on their knowledge of the reporting system. We note, that differences among users are very great, particularly with the absence of documentation.

In general, people find the report request routine hard to use. The report request option screen is also confusing. It contains a summary of the number of requests of different types of report without any explanation of what the information is. This does not pose any problem for the experienced user, but is a small nuisance and poses a problem for the new user. Another problem with the report request is when a "period" is given a transaction level on the accounting report, transaction detail is automatically added as the next report level without any warning. It is only the lack of warning and an understanding of the necessity for this practice that creates a problem. Finally, we note that the report request accepts more report definition levels than there are parameters in the options list.

In terms of dates, we observed that in general accounting, "period" on the screen becomes "effective date" on the report.

In addition, we found that not all information which is keyed in is accessible to reports. We observed this in a number of instances including:

- Accounting reports cannot access common object codes by patrol and crew.

- In accounts payable, two lines of type are accepted for a name, but only one is printed out on the cheque.

- In branch transfers, the transaction type is keyed in but doesn't appear on the printout.

- Information keyed in as "particulars" is not available on reports for accounts payable or maintenance management.

- Significant information is sometimes not available on the proof listing, for example, credit designation for dollar amounts on the monthly maintenance report.

The most important of these inconsistencies is that the whole report request procedure confuses its users. They do not understand what they can expect when they make a particular report request.

Information keyed in and not available is extremely important. It annoys people in your local offices that they can enter information and not be able to retrieve their own information when they want it. As well, it creates large possibilities for accounting errors.

Redundancy

Redundancy refers to system constraints which add unnecessary keystrokes to the operator's job. There are several types of redundancy that we have identified including:

- information common to all transactions in a batch which is entered for each transaction,

- long strings of information which could be replaced by a shortened code,

- repetition of information without any known purpose,

- other reasons for excessive keystrokes.

The redundancies identified are summarized in Table 2.

The most serious redundancy occurs when each transaction in a batch requires an item of information and the same information is required for each transaction throughout the batch. This occurs in many instances, including:

- effective date,
- transaction type.

Repetitions of common information are the most annoying type of redundancy to operators. They are also the most consistently costly in terms of wasted keying. Second to this in impact are the use of long strings of information when a shorter code would do the same job. Long strings tend to be more error prone and take up a considerable portion of operator time. The most serious instances of this type of redundancy are found in recording names and addresses -- particularly in accounts payable. One indicator of the impact of entering these long strings was found when looking at errors detected in batch proofs prior to transmission. About 28% of the detectable errors were due to keying errors in a name or an address field. To a lesser degree, use of a 10-digit vehicle code also seemed excessive**. Operators suggest that a shorter code would serve equally well.

Information which is entered twice with no obvious purpose is another source of annoyance and wasted keying. Instances we observed include:

- Municipal subsidies: reference number is entered twice.
- Invoices: reference number is entered twice.

**This is beyond the OMS system control. It is required by the equipment costing system.

- Advice of journal entry: must keep repeating particulars when they apply to a whole item.

Excessive use of the enter key to bypass fields is another form of redundancy. This is found in:

- The operator/crew/patrol/highway field on the advice of journal entry is skipped in most applications.

- In accounts receivable, after the name field, there are too many fields to skip before the next needed item.

- For funding work orders, you are required to hit the "enter" key about 10 times to get to totals on the cost centre budget form.

There are several other major sources of redundancy. By and large, users find the following things annoying:

- Having to skip the four spaces for initials before names are entered on invoices. Initials are only needed on about 10% of cases.

- Maximum of 10 invoices per company before payment -- this is often exceeded and results in a need to re-enter basic information.

We have identified other redundancies which trouble people considerably less -- they are of relatively minor importance but worthy of note. These include:

- When one copy of a report is requested it is always local, yet the location for the report is still requested by the system.

- Maintenance management -- credit and debit cannot be entered on the same form. You must use two forms if you have both to enter.

- Command keys are often multiple sequences, for example a shift asterisk for exit or shift and reverse arrow head for backspace.

Of the redundancies we have identified, we think there are two things which are of particularly high impact in terms of time consumed, errors produced, and operator annoyance. These two problems requiring immediate attention include:

- repetition of identical information in every transaction of a batch,
- typing of names and addresses.

Compared to these problems, all others have a relatively minor impact. As well, they produce considerably less irritation.

Error-Handling

In this section on error-handling we deal with six issues:

- error correction during data entry,
- error checks during data entry -- local edit,
- error correction using the MODIFY routine,
- constraints to error detection/correction,
- the use of beeps to signal errors,
- error messages.

Error-handling problems are summarized in Table 3.

Error correction during data entry is very limited. The only way you can correct errors is to backspace through previous items and re-enter. As you backspace to an earlier item, you erase everything along the path. In many instances, you cannot back up to previous items on the same screen. In these instances when people spot an error, they delete the entire transaction and re-enter it. This happened very often in our observations -- especially on maintenance management entries.

One of the characteristics of the Datapoint system is the field validation. Despite the large number of edits already available, some significant validations seem absent. These capabilities have a significant impact on batch rejection and on errors. Most significant is a missing validation for the compatibility of operation types with prefix/highway/-crew types. Lack of valid combination of these codes occurs in 24% of rejected batches. Thus, it has a substantial impact on system operation. Other validations which would be useful, include:

- Account number/cost centre compatibility -- this is present in 15% of rejected batches.

- Operator/crew/patrol/highway errors are detected prior to transmission in 9% of batches with pre-transmission detected errors.

- Equipment number and fuel type (gas, diesel) compatibility creates some batch rejections.

- Contract number-project number incompatibilities.

- Equipment work order batches are often rejected with recent equipment, because the equipment number is not yet entered centrally.

- In funding work orders, it is reportedly difficult to avoid invalid combinations of information. No specific fields were cited.

In addition to these problems which occur at data entry, there are many problems which occur in the MODIFY routine. This is the routine used to correct batches already entered. Problems observed include:

- Proof listings must often be requested instead of displaying a transaction using MODIFY, because it is so difficult to search through transactions in the MODIFY routine.

- Many batches have forms which are modified by citing the form number. To do this, the error must be spotted at the time of data entry and the form number manually recorded -- otherwise it creates considerable difficulty for access.

- You can only add or delete a transaction in a batch. You cannot add a line to a transaction or delete a line from a transaction. Yet, this is the commonest need. It is usually induced by skipping a line or duplicating a line during data entry.

- When you ask to modify a transaction, it is shown on the screen. When you enter the field number you want to modify, the value disappears. In fact, on the counterfoil field, two independent fields vanish.

- Often only one or two letters need to be corrected in a field, yet the whole field must be re-entered. This is costly when some fields are up to 50 characters in length.

- Some fields can't be corrected by any means. This includes an incorrect prefix for an operation and operator numbers for advice of journal entry.

- Batch status is automatically changed to balanced after MODIFY. This makes correcting spelling errors exceedingly difficult.

As well, some fields are hard to modify, such as the account number in an advice of journal entry. To modify this you must change the "A" to a "P", then change the project number, then change the "P" to an "A".

In addition to modifying most regular input forms, there is also the problem of modifying report requests. This is complicated by the fact that there is no list of report titles and numbers which would help you to rapidly identify the report request you may want to correct. Instead, you must keep track of this manually. Modifying a report request has several problems including:

- There is no listing of the parameters used in the report request -- you must search for them on a hit or miss basis if you have not manually recorded the details.

- When modifying a report request, you can't display a parameter value without being committed to a particular kind of change.

The use of warning signals or "beeps" is also an issue in error-handling. Nominally, beeps are only meant to signal errors. But, some beeps are created in fields which can be bypassed by the operator (e.g., in garage work orders, when the vehicle class, only, is known). As well, beeps are also used as warnings which are not errors -- this occurs too often. Some beeps are also used to call operator attention to something on the screen, such as the dollar charge for kilometrage on expense account forms. The cumulative effect of these beeps which do not signal errors is that operators may tend to ignore the beeps in some instances. Many

operators are also bothered by the multiple beeps which occur when an operator is typing well ahead of edit checks and an error is detected some way back. This not only disturbs the operator but other people around them.

The problem of errors being detected well behind where a person is currently typing is an annoyance to expert operators. Sometimes, in fact, the error is that the operator has typed in more information than the buffer waiting for validation can hold. This is a problem because:

- There is no provision for special error-handling when the buffer contains information beyond the current error.
- There is no warning or lock-out when a typist is about to exceed the buffer limits.
- All information is lost after the error field.

There are other problems in error-handling which are worthy of note. In particular, there are some messages which are unclear. Another issue is that an accidental switch to lower case makes it impossible to input information, yet no error message is produced.

Further, we observed that:

- Some error messages occur in fields which are bypassable.
- It is difficult or impossible to interrupt or back-up a processing activity.
- There is limited use of abort/confirm sequences.
- Some information is easily accepted yet results in central OMS errors (e.g., stock advice numbers).
- Correcting errors which involve calculations from tables, after the table has been updated, is extraordinarily difficult.

Of all the difficulties in error-handling mentioned in this section, we consider the most serious problems to be:

- It is impossible to correct all information visible on the current screen.
- Backing up to correct cannot be done without loss of input information and retyping.
- You cannot add a line to or delete a line from a transaction.
- Some fields cannot be modified.
- There are incompatibilities in operation type with patrol/highway/crew type in rejected batches.
- There is incompatibility between account number and cost centre.

We deem these errors to be most significant. But, there are many other errors that have a significant impact upon the operation of the system and especially upon the satisfaction of operators with the system.

Other Inefficiencies

A number of other problems we identified do not fall conveniently into any of the three categories we have already discussed. These tend to be general complaints about how the system is working or specific complaints about some of the routines. We have divided our discussion of these issues into two groups: applications-oriented and system-oriented. Both sets of problems are summarized in Table 4.

Applications-Oriented Inefficiencies

The majority of applications-oriented inefficiencies deal with problems in getting the reports that people want. There is a general feeling that report formats are not flexible enough -- that it is not easy to modify the way in which information comes out. Alternatively, some people believe the system may be flexible, but they lack the information on how

to use the report generator to its best purpose. Some supporting documentation with examples is necessary for the report request routine. Aside from this general complaint about report generation, people have made a number of observations about things which bother them, including:

- Cost centre reports are printed without page separation between cost centres. They must be physically cut in mid-page in order to be sent to the field.

- If a batch is rejected on the day of a report request, it never appears on a printout. The same problem occurs if a batch is held-over or entered late. This results in some batches never being present on a report.

- The lack of subtotals within batch proof listings make it more difficult to reconcile these listings with other documents.

- Many maintenance people would like to get a listing of results by common object code within a patrol/crew on the same report. This is not available.

- The maintenance transaction history report needs manual intervention before it can be sent out. Credits are not subtracted from costs on the report -- line or total. As well, one site reports that the top line of each page is missing and needs to be entered manually before sending it out (Hamilton).

In addition to these problems with reporting, we feel several points require another mention in this section including:

- Balancing batches by hash totals is dangerous, especially since the failure to note a credit is a very common error.

- The end of period for maintenance management requires a manual transfer of labour dollars held in a suspense account (61-employee benefits) into the proper accounts for these funds (60-for classified, 65-for unclassified).

- There is not enough space on the counterfoil for entering the needed information to identify hired equipment in accounts payable.

In addition, we must mention some complaints about an aspect of the field office work which is not part of OMS. That is, those batches which consist of all the transactions for a month, such as garage work orders. The offices do not like entering their data by sequence number and having a single batch for a month. It makes it impossible to do parts of the batch simultaneously using different people. Also, it forces them to generate phony batch totals each time they enter part of the batch -- even though they don't need them.

Of the problems stated here, the lack of understanding of report generation seems most critical. It is likely that reports are not being used to the full extent possible.

System-Oriented Inefficiencies

Users feel that the system is too slow especially for an expert keyer. Fast users get ahead of the system at all times. Also, when more than one user is on the system or the system is printing, even the slowest user must wait. Many offices have learned to avoid multiple users, but of course, this means that much of the equipment lies idle. The problem of slow transmission is aggravated for many operators who are expert, because they are typing well ahead of where the machine is editting. The keyboard is always active. Thus, when an error occurs for these operators, they lose a good portion of what they have entered. It is probable though, that this slowness cannot be dealt with so long as a field validation (edit) procedure is used.

Compounding the problem of slowness of data entry is the fact that the system is sometimes not available for use. The mainframe system is reported as being down often and not able to receive. As well, special transmissions (i.e., equipment costs) tie up the system for periods of time. The system is down from 20 to 30 minutes on transmission; then 30 to 60 minutes later it is shut-down for another 15 minutes to receive the report. Then, if everything is not perfect, the process must all be repeated again. In addition, operators are annoyed that their machine

becomes inoperable when a proof-listing has been requested from their keyboard.

Problems of transmission are another complaint. Many of these seem to have occurred in the summer of 1982, but the residual effects remain. People think that transmission should be more reliable. As well, most would like some documentation to indicate what to do if they run into transmission difficulties. While they indicate that good response to their problems is available from headquarters, most would like some kind of documentation available to take quick action on their own.

As it stands now, most people rely on informal notice and discussions with other people in their office to deal with system problems. When these fail, they resort to phone calls to Toronto.

Transmission is particularly a problem if the end-of-day transmission is interrupted. If this happens, manual intervention is required. The operator must manually reset the date of the last information request in each report request and must alter each and every status indicator for every batch being transmitted that day.

Finally, as a result of previous transmission difficulties, many districts would like some way to back-up their data locally, so as not to tie up all the memory on their system. A few districts have not had sufficient memory left to do their day's work because of failure to transmit on successive days.

As well as the problem of storage of data locally for backup, many sites believe that local staff need better access to their own database. They would like some way to get at all of their information. This is partly because many places believe that the turnaround on reports is slow. Often they need information quicker than they can get it. This is particularly true if there has been a batch rejection on the day that a report is required. This kind of thing can impose as much as a 48- to 72-hour delay on getting a report out.

In addition to problems of slowness and delays, users have several complaints including:

- There is a general lack of documentation to support system operations.

- Clerks must get managers (or someone authorized) to release batches requiring correction even though they have not yet been transmitted.* This poses a delay which they believe is unnecessary.

- It is annoying to prepare the annual report for a centre by hand, when most of the information is already on the system. They would like to see the annual report completely produced by the system.

- People claim they are often not told why a procedural change is needed. They would like to understand the reasons behind requested actions.

- There is too much reliance on memory or manual recording for some important information. There is no easily accessible list of report request titles and numbers. Also, you must remember the transaction number of an item with an error in a batch in order to correct it. There is no easy way to page through the batch instead.

- Batch rejection slows down the work tremendously. Many sites wonder why a particular line of code can't be rejected from a batch and the remainder of the batch accepted.

- Screen problems are reported at some sites. The problems include flutter, random letter expansion, and other idiosyncrasies.

Aside from these difficulties, we made a few observations of our own.

- There are no help commands available once somebody is stuck in a routine.

- Only one type of dialogue is available, menu plus question and answer. There is no expert user mode for operating the system.

---

*This difficulty is intentional -- part of audit requirements.

- Menu selection is a strict up-down hierarchy. There are no sideways jumps possible.

Of the problems mentioned here, the most serious are:

- the slowness of data entry,
- transmission problems.

APPENDIX D/ SOLUTIONS

Changes in Data Entry

There are five changes in data entry procedures which we consider imperative (see Table 5). First, information which is repeated on every line of a batch should be moved to the batch header. This includes such information as date, transaction type, and activity code. This should result in more efficient data entry and would remove one major source of operator dissatisfaction with the system.

Second, validation should be done on entire lines of data (or logical groupings), rather than on individual fields. That is, each field should still be validated; but, all the fields on a line should be checked after the entire line is entered. This, combined with some later suggestions for error correction, should make correction easier for users. As well, by reducing the number of switches between terminal and processor, the speed of data entry should be improved.

A third change of major importance is to increase field sizes in some of the maintenance transactions. In particular, hours worked should be allowed to exceed 1000 and the litres of fuel consumed should be allowed to exceed 10 000. This would have a substantial impact on reducing errors and time of data entry. It would remove the need for notional transactions which are a major burden to users entering maintenance reports.

Finally, adding two local edits to the system would have a substantial impact on reducing batch rejections and thus would improve turnaround time for reports and reduce the need for manual intervention to correct errors. First, there should be a local compatibility check on the patrol/highway/crew type combination for each operation type (Error 2161). Errors of this type occur in 24% of rejected batches. Thus, a compatibility check could decrease the number of rejected batches. The second useful compatibility check is between cost centres and associated account numbers. This accounts for 15% of batch rejections and errors detected prior to transmission.

Once we leave the realm of high impact recommendations, the number of changes possible multiplies quickly. Some of these changes may not be necessary if other changes suggested are implemented. Nonetheless, they are presented here. The list of recommended changes here is a long one. It includes:

- Any field which contains a default value should be bypassable by pushing the enter key once. Where there are multiple fields which must either all be filled or all be absent, a single enter key should bypass all the fields. Some examples where this happens are between name and next needed information in accounts receivable; to move from line items to total on funding work orders; to skip over the operation, crew, highway fields on an advice of job entry; and to skip over the initials in the name field on accounts payable.

- Data entry should only move forward by pressing the enter key. There should be no automatic moves.

- Add a local validation on the compatibility of contract numbers and project numbers.

- Allow multi-purpose trips for reporting maintenance activities.

- When examining batch balances, use net dollar amounts as well as hash totals.

- For accounts payable, increase the number of invoices per payment or allow an easy copy of header information to the next set of 10 invoices.

- Eliminate or reduce the number of entries for twice entered items such as reference numbers on invoices and municipal subsidies; or particulars for an advice of journal entry.

In addition to these changes, some method is required for dealing with information stored in a buffer when a user has typed well ahead of the local validations and an incompatibilty is detected in a previous field. This is a problem encountered in data entry, it is discussed in the

section on improving operator flexibility. All we would suggest here, is that if a user has completely filled the buffer, the keyboard should be locked and a "beep" sounded to indicate that the buffer is full.

Further to the aforementioned suggestions, there are many other changes which would primarily improve the satisfaction of system users. Recommendations in this class include:

- Screens: revise source documents or screens to achieve consistency, depending on which is less expensive to revise.

- Report requests: review the report request screen to remove excess information. Order report request options in a logical fashion. Have an on-line "help" command to explain the options.

- Cost centre reports: start a new page for each new cost centre. As well, the cost centre number should be printed on the top of each page. This would help offices in transmitting the reports to their cost centres.

- Menus: alter the menus so that options are re-ordered by frequency. Also, indicate which menu items are not yet developed. Finally, standardize the position for entering menu choices, the terminology directing menu choices and the form of the data for specifying menu choices.

- Beeps: use "beeps" to signal errors only. Remove beeps used for warnings or to get operator attention. Limit the number of beeps to one at a time -- remove long sequences of beeps resulting from reading through the buffer.

- Shorten vehicle codes.

- Allow both credits and debits to be entered in a single maintenance report.

- Invalid data should remain visible until corrected.

- Error messages should remain visible until corrective action is completed.

- Provide sub-totals at appropriate locations in proof listings.

We have presented these "grocery lists" of recommendations at a relatively general level. We have done this to avoid repeating details which are cited in Appendix C. But, for those who want a cross-reference between suggestions and problems, this cross-reference will be presented in Appendix E.

Improve Operator Flexibility

In general, recommendations made in this section will probably be more difficult to implement than those in the preceding section. They tend to have wider ramifications on how operators can work with the system. Most of the suggestions arise out of problems in handling errors. All of the suggestions in this section are summarized in Table 6.

For error-handling, the single most important change you can make is to replace the backspace and erase procedure for correcting errors. Instead, the user should be able to move the cursor to any place on the screen without erasing the material between the position of the cursor when an error is found and the position of the error itself. Then, the user should be able to overstrike any characters to be changed. Subsequently, the user should be able to move the cursor back to its original position and continue entry without any penalties. Accompanying this free cursor movement and overstrike capability, the user should be able to alter everything in the batch. No field should be inaccessible. By increasing the ability to correct errors during data entry, less use of the MODIFY routine will be required.

Another recommendation with high impact is that the MODIFY routine allow users to add or delete any line from any transaction or form. It should not be necessary to remove a transaction entirely or add an entire new

transaction to make this kind of alteration. This will improve the efficiency of data entry and modification considerably.

Error-handling would also be facilitated if there was a method for scanning the transactions in a batch -- especially in the MODIFY routine. A similar capability during data entry, combined with the capability of altering anything on the current screen, might eliminate the need for a MODIFY routine.

A number of suggestions with a lesser impact also deal with error-handling. These include:

- Use overstrike to correct fields in the MODIFY routine -- do not have the previous entry in the field disappear.

- Make all fields modifiable.

- Provide some facility for modifying "tabled" information when making a correction to a batch from a previous day.

Many of the suggestions we have made also deal with the problem of errors detected when an operator is typing well ahead of the edit. The optimal suggestion, in our view, is that when an error is detected the keyboard is frozen. The operator is allowed to return to the field with the error without losing any characters. Then the operator can insert characters in that field and/or move information in the buffer into that field. Subsequently, the user should be able to return to where they were when the error was detected. This procedure would allow for an easy correction and also prevent the loss of information already available in the buffer. If this were impossible, then we would recommend a procedure which is almost diametrically opposed: freeze the keyboard when an error is detected, return the user to the error field, and delete everything in the buffer. This would prevent the multiple "beeps" which annoy users and also speed up the process of error correction.

Aside from error-handling, there are only four recommendations for improving operator flexibility:

- All information entered should be accessible to reports.

- The method of exiting from any routine should be standardized -- preferably with a single function key. This key should also be usable during any processing -- it should permit interruption of processing.

- Two-stroke functions, such as shift plus asterisk or shift plus < should be replaced with one-stroke functions, if possible.

- A single reset command should be available for returning information to the pre-transmission state when an end-of-day transmission is interrupted.

Add Files

There are only two files which we would suggest adding to the system (see Table 7).

The highest impact suggestion we can make is to add a vendor file to the system using telephone numbers to replace names and addresses of vendors which are commonly entered by users. The use of a vendor code, instead of names and addresses, could be optional. You might restrict vendor codes to province-wide entities like Bell Canada or Ontario Hydro. Alternatively, and this is our long-term suggestion, it could be used for all regularly invoiced corporations and for all expense accounts. Vendor code files would eliminate a large part of data entry time. They would also eliminate a major source of errors found prior to transmission and errors found in batch rejections. Aside from eliminating information which is repeated on every line, this is probably the second most popular suggestion made by users.

The vendor code file is certainly the only high impact file that can be added. In addition, however, we would also suggest adding a local file containing information on reports which have been requested. This file would contain the report number, title, parameters requested, and the values of each parameter requested in the report. You could also make this file valuable by working it into a routine which displayed a quick schematic of what the report would look like. This approach might

relieve the need for more elaborate documentation of the report generation system.

Other Issues Requiring Attention

The remaining issues (see Table 7) requiring attention can be divided into questions of documentation and questions of revising the system. In terms of documentation, users are nearly unanimous in their desire for better documentation of the report generation system. There has been some question of the flexibility of the system, but we believe this may be due to a lack of understanding as to how it works. Documentation of the report generation routine is a high priority. Note though, that the documentation should include many examples of what would appear from a set of parameters.

A second area requiring documentation is transmission problems. Most sites would like a basic manual indicating what to do about some commonly occurring transmission problems. While they recognize that good support is available at headquarters, they would prefer to deal with their problems systematically and on-site.

Finally, a few sites indicated that they would like some documentation for training. This is a minority request in our experience, since most users have reported little difficulty in mastering the system -- even without documentation. We suspect that such a manual used as a supplement to personal training, might improve mastery of the system. But, we think it merits a low priority.

In addition to documentation, some users complained about having to manually produce annual reports. User goodwill could be gained if the annual report of each centre could be produced from the OMS. Users believe that all information, except county and riding, is currently on the system and resent having to piece it together manually when it is already there. They suggest that a table showing how to allocate costs by county and riding could be added so that annual reports could be generated by the system.

Finally, there are a number of issues which we consider to be system-based. These are issues which we can identify, but remedying them will probably prove difficult. These issues include:

- line rejection rather than batch rejection,
- provide a local back-up for data,
- increase the speed of data entry,
- increase transmission reliability.

APPENDIX E/ SUMMARY OF PROBLEMS AND SOLUTIONS

This section provides a brief overview of the findings. Then, problems and suggestions for correcting problems are tied together in a series of tables.

Overview

Users are generally satisfied with the system and think it is a good system. They find the system very easy to learn and easy to use. Their complaints reflect a desire to see a good system become even better, rather than a rejection of the whole system.

The major problems that users commented on were:

- Some information is repeated on every transaction within a batch and doesn't change throughout the batch.
- The same names and addresses are usually entered each month. These take a long time to enter and are very error-prone.
- Fields for entering hours of work and litres of fuel are too small on maintenance management reports.
- Lack of certain validations lead to excessive batch rejection.
- Lines cannot be added or deleted from a transaction in the MODIFY routine.
- Backspacing to an error during data entry erases all of the good information along the way.
- Some information cannot be corrected -- easily or at all.
- There is no way to scan the transactions in a batch easily.
- All information keyed-in is not available for reports.
- Batch rejection slows down work too much.
- The system is too slow for data entry.
- Transmission is sometimes not reliable.
- The report request routine is confusing and very hard to use.

To correct these difficulties and others, we have made a number of suggestions which we think will have a significant impact on system efficiency. These recommendations include:

- Provide free cursor movement and an overstrike capability for error-handling.
- All information in a batch should be correctable during data entry and MODIFY.
- Switch validation from field validation to line validation.
- Any information which must be entered on every line in a batch should be moved to the batch header and eliminated from the line.
- Add recommended local validations of operation/patrol/highway/crew and account number/cost centre.
- Increase the field sizes for hours and litres in the relevant screens.
- Allow addition and deletion of lines from a transaction in the MODIFY routine.
- Provide a facility for scanning all entries in a batch.
- Improve the documentation on the report generation system.
- Add vendor account files to replace names and addresses.

Relationship Between Problems and Solutions

In this section we attempt to briefly describe the relationship between problems and solutions. Rather than explaining this in words, which would be very cumbersome; we present the information in a series of tables.

The tables in this section are organized in a way which corresponds to the tables which list problems. Table 8 identifies solutions to the problems identified in Table 1 -- inconsistencies. Table 9 identifies solutions to the problems listed in Table 2 -- redundancies. Table 10 lists solutions to problems presented in Table 3 -- Error-handling. Finally, Table 11 presents the solutions to the unclassified problems presented in Table 4.

In each table the problems are identified by a number. The number corresponds to the problem number associated with the verbal description of the problem in each of the problem identification tables. Opposite the problem number is the solution number. The solution number consists of two parts: a table number and a solution number. The number preceding the dash is the table number. This identifies the table where the solution description can be found. The number following the dash is

the number which follows the verbal description of the solution in that table.

Thus, the tables in this section cannot be understood without referring to the tables in Appendices C and D. While this method is cumbersome, it seems less cumbersome than to provide elaborate verbal descriptions of the relationships. This is particularly true, since most of the relationships are obvious to an informed reader.

Table 1/ Inconsistencies

System	1/ Options on the menu are not always in production.
	2/ Order of menu options does not reflect frequency of use or logic.
	3/ No standard entry window for entry of menu selections.
	4/ Some main menu applications lack an intermediate options screen.
	5/ Terminology describing menu choices is inconsistent.
	6/ Different procedures for exiting from routines.
	7/ Inconsistent defaults in data entry.
	8/ Differing treatment of invalid data fields.
	9/ Error messages sometimes vanish before any action can be taken.
	10/ Some by-passed fields require an entry.
	11/ Data definition of menu choice entry is inconsistent.
Form	12/ Lack of common style to screen layouts.
	13/ Data entry moves forward in different ways -- automatic and manual.
	14/ Operation codes do not match the way work is done.
	15/ Data entry source documents often differ from screens in some particulars.
	16/ Total dollar amount hash is used instead of total dollar amount net.
	17/ Fields for hours and litres are too small on the maintenance management, garage work order and equipment cost screens.
Report	18/ Report request routine is hard to use.
	19/ Report request screen contains unidentified and poorly formatted information about the number of reports requested.
	20/ Report parameters are entered automatically in some instances.
	21/ All information keyed-in is not accessible in reports.

Table 2/ Redundancies

1/ Effective date and transaction type are repeated for each transaction in many batches and usually do not differ throughout.

2/ The same names and addresses are usually entered each month. These take a long time to enter and are very error prone.

3/ Users find the length of the vehicle code (10 digits) excessive.

4/ Some information is entered twice without any error or consistency check between the two entries.

5/ Some batches demand excessive use of the enter key to bypass unused fields.

6/ Skipping four spaces for initials is annoying and causes errors. Fewer than 10% of entries require initials.

7/ Limit of ten invoices per payment is too low.

8/ Some transactions require two forms to enter.

9/ Command keys are often multiple sequences.

TABLE 3/ Error-Handling Problems

1/ Backspacing to an error during data entry erases all of the good information along the way.

2/ Some items on the current screen cannot be corrected during data entry.

3/ Some significant local field validations are missing or not functioning properly: prefix/highway/crew/ operation, account number/cost centre and others.

4/ Previous screens cannot be scanned or corrected during data entry.

5/ There is no way to scan the transaction in a batch easily.

6/ A line cannot be added or subtracted from a transaction -- you must add or delete the entire record.

7/ Fields disappear when a change is requested in the MODIFY routine.

8/ A whole field must be re-entered, even though only one character may need correction.

9/ Some fields can't be corrected by any means.

10/ Some fields are excessively hard to modify.

11/ Batch status changes automatically after MODIFY.

12/ No listing of report requests or parameters is available to facilitate changes.

13/ "Beeps" are used to signal too many things.

14/ The system has no special provision for dealing with errors detected behind the operator's current position. There is no facility for dealing with data stored in the buffer.

15/ Some error messages are unclear and occur when not needed.

16/ Difficult to interrupt processing and back-up.

17/ Difficult to correct errors which involve tables that have been updated since data entry.

TABLE 4/ Problems Not Elsewhere Classified

Applications	1/ Cost centre reports do not start each cost centre on a new page.
	2/ A batch rejected on the day of a report request never appears on a printout.
	3/ Lack of sub-totals in batch proof listings hinder reconciliation.
	4/ Some reports desired cannot be printed, even though the information is on the system.
	5/ Maintenance transaction history requires manual intervention before distribution.
	6/ Hash totals are a nuisance compared to net totals.
	7/ Counterfoils sometimes lack the space needed for identifying information.
System	8/ The system is too slow.
	9/ Transmission is sometimes not reliable.
	10/ Equipment cost reports tie up the system during the the middle of the day.
	11/ Interruption of the end of day transmission imposes severe penalties on users.
	12/ Districts have no local storage for backing-up data.
	13/ Lack of system documentation in general, especially for report generation and transmission problems.
	14/ Annual reports must be manually produced.
	15/ Justification for system procedure changes is sometimes not given.
	16/ Too much reliance on memory for important information.
	17/ Batch rejection slows down work too much.
	18/ Some screen problems.
	19/ No help commands available.
	20/ Only one type of dialogue available -- no expert user mode.
	21/ Menu selection is strict up-down.

TABLE 5/ Suggested Changes in Data Entry and Report

Large System Impact	1/ Information repeated for every transaction in a batch should be moved to the batch header.
	2/ Increase field sizes for hours worked and litres of fuel.
	3/ Add a local validation for operation type with prefix/ highway/crew.
	4/ Add a local validation for account number/cost centre.
	5/ Validation should be changed from field validation to line (or logical grouping) validation.
Moderate System Impact	6/ Any field(s) which contain a default value should be bypassable by pushing the enter key once.
	7/ Data entry should move forward by operator action only.
	8/ Add a local validation of contract numbers/project numbers.
	9/ Allow multi-purpose trips for reporting maintenance activities.
	10/ Use net dollar amount totals as well as hash totals.
	11/ For accounts payable, increase the number of invoices per payment.
	12/ Eliminate information entered twice unless there is a validation check.
	13/ Provide some facility for dealing with information stored in the buffer when an entry error is detected.
Impact on User Satis-faction	14/ Revise source documents/screens to achieve consistency of layout.
	15/ Redesign the report request screen.
	16/ Have an on-line "help" command to explain report request options.
	17/ Start each cost centre on a new page in a report.
	18/ Re-order menu options by frequency.
	19/ Standardize menu layout and choice options.
	20/ Use one "beep" to signal an error sequence. No beeps for any reasons other than errors.

21/ Shorten vehicle codes.

22/ Allow credits and debits to be entered in a single maintenance report.

23/ Invalid data should remain visible until corrected.

24/ Error messages should remain visible until corrective action is completed.

25/ Provide sub-totals at appropriate locations in proof listings.

TABLE 6/ Suggested Changes for Improving Operator Flexibility

High System Impact	1/ Free cursor movement with an overstrike capability for correcting errors in data entry.
	2/ Allow all information in a batch to be corrected during data entry.
	3/ Lines should be capable of being added or deleted from a transaction in the MODIFY routine.
	4/ Add a facility for scanning a batch.
Moderate System Impact	5/ Use ovestrike to correct fields in the MODIFY routine.
	6/ Make all fields modifiable.
	7/ Provide a facility for modifying "tabled" information in batches corrected after a table update.
	8/ Provide a facility for dealing with information stored in the buffer when an error is detected.
Impact Primarily on User Satisfaction	9/ All information entered should be accessible to reports.
	10/ Exit procedures should be standardized.
	11/ Two-stroke functions should be reduced to one-stroke.
	12/ A single reset command should be available for returning information to the pretransmission state when an end-of-day transmission is interrupted.
	13/ Batch status should be changed to "balanced" and "transmission-ready" only with user agreement -- not automatically.

Table 7/ Other Suggested Changes

High System Impact	1/ Replace names and addresses with vendor account codes. Use telephone numbers for the codes.
	2/ Provide better documentation for report generation.
Moderate System Impact	3/ Provide documentation for dealing with transmission problems.
	4/ Line rejection rather than batch rejection.
	5/ Increase the speed of data entry.
	6/ Increase transmission reliability.
Impact on User Satisfaction	7/ Add a report request file including report number, title, parameters requested and parameter values. Generate a sample report layout from this information.
	8/ Provide some documentation to support training.
	9/ Have the system produce annual reports.
	10/ Provide a local back-up for data.
	11/ Error messages should be user-oriented and point to the action needed to correct an error. They should only occur when corrective action is needed.
	12/ All reports should be produced by information transmitted at the end of day. They should all be produced with the start-of-day transmission.
	13/ Users should receive justificaiton for procedural changes, preferably via the start-of-day transmission.

Table 8/ Solutions to Inconsistencies in Table 1

Problem Number (Table 1)	Solution: Table and Number
1	5-19
2	5-18
3	5-19
4	5-19
5	5-19
6	5-10
7	5-6, 5-7
8	5-23
9	5-24
10	5-6
11	5-19
12	5-14
13	5-7
14	5-9
15	5-14
16	5-10
17	5-2*
18	5-15, 5-16, 7-2, 7-7
19	5-15
20	5-16, 7-2, 7-7
21	6-9

*Large System Impact

TABLE 9/ Solutions to Redundancies in Table 2

Problem Number (Table 2)	Solution: Table and Number
1	5-1*
2	7-1*
3	5-21
4	5-12
5	5-6
6	5-6, 7-1*
7	5-11
8	5-22
9	6-11

---

*Large System Impact

TABLE 10/ Solutions to Error-Handling Problems in Table 3

Problem Number (Table 3)	Solution: Table and Number
1	6-1*
2	6-2*
3	5-3*, 5-4*, 5-8
4	6-2*, 6-4*
5	6-2*, 6-4*
6	6-3*
7	6-5
8	6-1*, 6-5
9	6-6
10	6-6, 6-7
11	6-13
12	7-7
13	5-20
14	5-13, 6-8
15	7-11
16	6-10
17	5-7

*Large system impact

TABLE 11/ Solutions to the Unclassified Problems in Table 4

Problem Number (Table 4)	Solution: Table and Number
1	5-17
2	7-4
3	5-25
4	5-16, 6-9 7-2*, 7-7
5	5-17, 6-9
6	5-10
7	-
8	5-5, 7-5
9	7-3, 7-6, 7-10
10	7-12
11	6-12
12	7-10
13	7-2*, 7-3, 7-8
14	7-9
15	7-13
16	6-4*, 6-10, 7-2*, 7-7
17	7-4
18	-
19	5-21, 7-2*, 7-8
20	-
21	-

*Large system impact